Price $1.50

# SCCS INTERFACE

VOL 1 ISSUE 6 MAY 1976

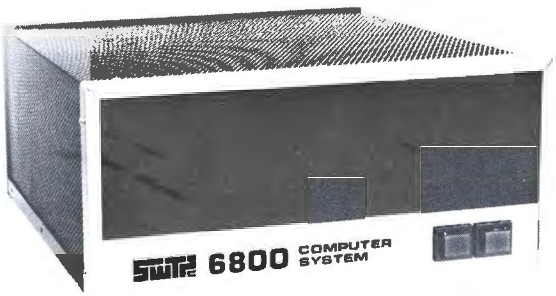
PUBLISHED FOR THE HOME COMPUTERIST


THE NATIONAL PUBLICATION OF THE SOUTHERN CALIFORNIA COMPUTER SOCIETY

# COMPARE

| Feature | SWTPC –6800 | Theirs—Your Choice |
|---|---|---|
| Processor— | The best "Motorola MC6800". Two accumulators, automatic vectoring, seven addressing modes and complete set of branch instruction. The more powerful instruction set and memory orientated architecture makes programming very straight forward and easy to learn. Operates from a single +5 Volt supply. | Some are almost as good in one respect or another. None can offer all of the features of a real MC6800. |
| Memory— | Static 2102-1 type memories. Fast enough to allow the processor to run at full speed at all times. No refresh cycles, no problems with glitches and flakey dynamic memories. | Various types available. Often not included in the basic kit, and must be purchased as an extra cost option. (this is an option?) |
| Power Supply— | 10 Amp. Capacity. More than enough to power a fully expanded system. Power supply uses a rugged 25 amp bridge rectifier and a 91,000 mfd computer grade filter. Regulators on the individual plug-in cards. | Some expand more than others with the supply provided. Check carefully. |
| Expansion— | Seven slots for processor and memory boards. Eight I/O slots. I/O's are programmable type. All decoding and clocking provided from mother board making additional interfaces very inexpensive. Baud rates may be independently selected for each interface card. | Varies from "0" to 16, or more. |
| Start Up— | Automatic start and reset provided by "Motorola" Mikbug® ROM. No fiddling with switches and status lights. Just push the button and go. Use of standard Motorola firmware makes software 100% compatible with Motorola evaluation module programs. | Anything from switch and status light to automatic ROM loading. If ROM is not a standard part, the software may be unique to that machine. |
| Clock— | Crystal controlled master clock oscillator with high power clock drivers. Insures reliable, consistant operation with no noise problems. Baud rate divider operating from the master clock oscillator provides the various baud rates for the I/O devices with crystal accuracy. No adjustments necessary to lock everything in at the proper frequency. | Anything from cheap dual monostable systems to crystal control. Crystal oscillators are best. Dual one-shots can develop phase overlap problems and are more susceptible to noise problems. |
| Buffering— | Tri-state bi-directional buffers on all data lines, address lines and clock lines on ALL boards. Insures trouble free noise immune operation. | Various—from full buffering to almost no buffering. Lack of full buffering can lead to noise problems. |
| Documentation— | Very complete. Our own notebook, plus the "Motorola" Programming Manual and Applications Manual. Detailed instructions along with sample programs help you to understand programming. You will be ready and able to write your own programs after mastering these instructions. | |
| Price— | $395.00 For the whole thing. You get the case, power supply, processor 2K word memory and serial interface. No extras to buy. Just connect a terminal and start operating. | As we said—COMPARE. |

® Registered Trademark Motorola




## COMING ATTRACTIONS

CASSETTE INTERFACE—Our new AC-30 will make it possible to load and dump programs to cassette tape using ordinary inexpensive recorders. Uses the standard "Kansas City" recording format for compatibility.

LINE PRINTER—At last—hard copy at a reasonable price. Our new PR-40 printer will print program listings, or any other material you might want to keep in printed form. A dot matrix printer at a price you are not going to believe.

GRAPHICS TERMINAL—A universal—works with any computer—graphics terminal. With this connected to your computer you can play games in style. May be used with our CT-1024 to put both graphics and alphanumerics on the screen simultaneously.

PLEASE—Don't call or write. We will have details on these projects in our next ad. Government regulations prevent us from giving prices, or taking orders yet.

SOFTWARE—The flood is near. Editor and assembler now available. BASIC and more games right away. Yours for the cost of copying. WE DON'T SELL SOFTWARE—WE GIVE IT TO YOU. ENJOY IT, COPY IT, WE WON'T COMPLAIN. . .

SWTPC Southwest Technical Products Corp., Box 32040, San Antonio, Texas 78284

CIRCLE NO. 1 ON INQUIRY CARD

sccs INTERFACE

The National Publication of the Southern California Computer Society

Vol. 1 Issue 6 MAY 1976



OBJECTIVES AND PUBLICATIONS

A non-profit organization whose purpose is:
A. To exchange and disseminate information among the Society members concerning the computer arts and sciences.
B. To provide a technical assistance to other members of the Society in those computer projects which are not undertaken for pecuniary gain or profit including but not limited to hardware, software, and computer programming.
C. To publish books, newsletters, magazines, and other periodicals for the benefit and education of the Society members and the general public.
D. To control and sponsor seminars, lectures, and courses relating to the computer arts and sciences.
F. To develop and maintain computer centers and laboratory workshops for the members of the Society and the general public including provisions for time-sharing operations.
SCCS INTERFACE, the official publication of the Southern California Computer Society is published monthly. Its content is composed primarily of articles contributed by members of the Society and intended for the reader with an interest in computers for professional or avocational reasons.

MEMBERSHIP

Membership in the Society is open to anyone, regardless of educational background or geographic location. All members are entitled to the rights of meetings, elections, receipt of official publications in a timely manner, and all other benefits as provided by the officers, Board of Directors, and various committees of the Society pursuant to the by-laws of the Southern California Computer Society.



EDITORIAL CONTRIBUTIONS

They must be accompanied by return postage and we assume no responsibility for loss or damage thereto. Any material accepted is subject to such revision as is necessary in our sole discretion to meet the requirements of this publication. The act of mailing a manuscript and/or material shall constitute an express warranty by the contributor that the material is original and in no way an infringement upon the rights of others. Send all contributions to: Art Childs—Editor, 335 North Adams, Suite 210, Glendale, Ca. 91206.

# CONTENTS

## Features



## Featurettes



## Departments



PUBLISHERS STATEMENT

All editorial content and effort are contributed at no cost to the Society. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the Southern California Computer Society. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes an endorsement of the product or products by this magazine or by the Southern California Computer Society.
Interface magazine is published monthly by McPheters, Wolfe & Jones, 6515 Sunset Blvd., Suite 202, Hollywood, Calif. 90028. Subscription rate $1.00 with membership in Southern California Computer Society.
Application to mail at second-class postage rates is pending at Los Angeles, California 90051 and at additional mailing offices.

**POSTMASTER:** Please send change of address form 3579 and undelivered copies to: Southern California Computer Society, P.O. Box 3123, Los Angeles, CA. 90051.

**DIRECT ALL ADVERTISING INQUIRIES TO:** Bruce Berkey, National Sales Manager, Advertising Department, SCCS INTERFACE, 61 South Lake Avenue, P.O. Box 4566, Pasadena, California 91106—(213) 795-7002. Steve Richards, Advertising Representative; Zack Bovinette, Store Circulation Manager.

*Art Childs, Editor*

*Photo by George R. Clarke*

# INTERFACIAL

Receipt of a letter from an author of the Altair Basic Interpreter prompts me to comment on a subject that is becoming the source of considerable controversy: PROPRIETARY SOFTWARE.

The letter I received complained that proceeds from sales of MITS BASIC had resulted in the equivalent of $2 per hour compensation for the writer's efforts, and implied that the obviously inadequate return on his investment for time and labor was because " ... most of you (hobbyists) steal your software."

Because of the letter's overall tone, I can't take those comments very seriously. I can take seriously, however, the problem pointed out by that letter: how are those who write software for the home computing market to be compensated for their efforts?

The answer to this crucial question would appear obvious—the simple exchange of money for the opportunity to use the software—much the same kind of transaction utilized to purchase a book on digital design techniques from your local electronics store. But a little thought quickly brings about a realization of the complexity of the problem.

The old adage, "possession is nine points of the law" expresses a philosophy quite adequate when dealing with tangible objects such as cars, houses and hardware. The applicability of such a concept to intangibles such as loyalty, ideas and software is difficult to imagine. Our culture has taken certain steps in the past to rectify problems in this area by instituting patent and copyright laws, and put considerable effort into enforcing those laws. But, the success of enforcing the intent of such impediments has been limited.

Consider the case of the technical book purchased at an agreed-upon price from the author. The paper, ink and binding materials to make up that book are certainly not worth the $10 to $20 paid by the end user. The ideas on the other hand, could be, and often are worth thousands of dollars. Everything considered, that which the publisher of the book really offered, and that which the author of the book really sold, was a particular set of concepts and ideas. Imagine how ridiculous the publisher would appear should he attempt to require the purchaser to refrain from communicating the concepts expressed within the book.

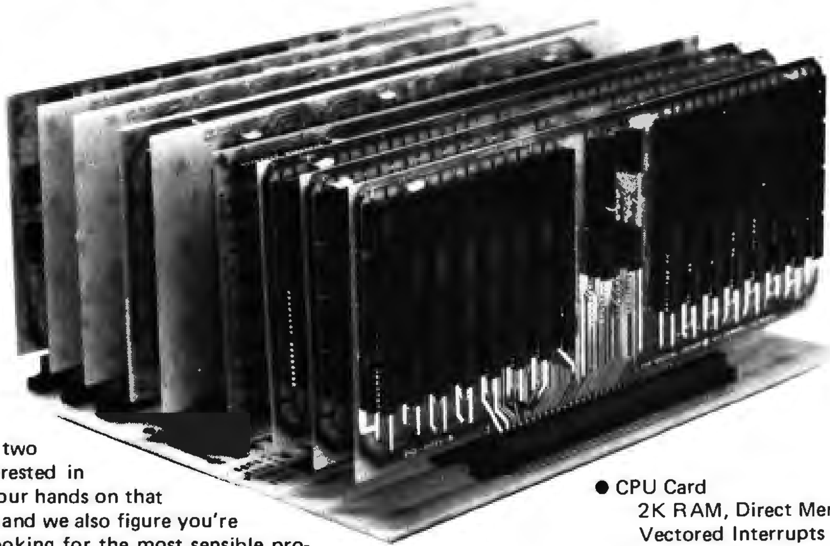
The fact that current laws prohibit certain kinds of communication of concepts in no way lessens the absurdity of the intent of those laws. Had the real intent of those laws been to restrict the use of the mechanics of communication of information, rather than the information itself, it would be a different ball game.

So what are the rules of this software ball game? Should it be illegal (or unethical or immoral) to give your friend the Basic Interpreter you purchased? And if so, exactly what action on your part would constitute an illegal (or unethical) act? The act of giving him a paper tape in object format—or the act of loaning him the tape (assuming he will load it into his machine and dump it from memory onto his own paper tape)? Or, to take things a step further, would it be improper to allow him to run Basic on your machine (improper because you, not he, were the one who paid to use the interpreter). Or perhaps we could split hairs and say it's okay for your friend to use Basic on your machine, but not his, even though you were there to assure that he did not make a copy for himself. Or to go a bit further, it might be permissible for him to use Basic on his machine provided he returned to you any output; i.e., listings, program listings, or data listihgs representing the results of the utilization of a program which he himself did not purchase.

Most people agree it would be wrong for the purchaser of the program to resell it. But should this be? Is it wrong to sell your car when you are through with it? Of course not! So why should it be wrong for you to sell a long strip of tape with a bunch of holes in it once you are through using it? The answer of course, is yes, it's wrong—not because you are selling a strip of paper, but wrong because you are selling a complex set of holes, spaced in such a way as to be interpreted according to a set of concepts, the majority of which were conceived at Dartmouth University, not Albuquerque, New Mexico. [To be complete and accurate, we must acknowledge that even those concepts conceived and implemented at Dartmouth University were built upon and utilized concepts and ideas that have been part of man's heritage since before the days of Euclid and Pythagoras.]

I don't pretend to have any conclusive answers to this software problem (a problem that is not confined to the hobbyist market), but I do believe this: Man's every attempt to legislate morality, has failed sooner or later. The only laws that prevail in the end are the laws of nature. In light of this, it seems to me that we have two choices: One, assure that the concepts which are embodied in a complex logic set cannot be stolen

# MEET THE DIGITAL GROUP



If you're taking time to read this ad, we already know a thing or two about you. We know you're interested in devouring everything you can get your hands on that deals with microcomputer systems, and we also figure you're out there shopping around . . . looking for the most sensible products to fit your needs and your budget.

That's why we think you should get to know us.

We're the Digital Group, a relatively small, 18-month-old organization obstinately dedicated to providing quality in every product we offer. You may have already heard a little something about us from a friend — our reputation does seem to be getting around quickly, even though we've never advertised before.

We think it's due to a number of important factors: state-of-the-art designs, a really complete systems philosophy, unexcelled quality, reasonable software, three-week delivery, and no pre-announcements until we're ready to deliver. Our products are not just a gleam in our designers' eyes; they are currently being delivered . . . fast!

**The Advantages**

Here are a few specific advantages of our product line:

- We offer CPU's from different manufacturers which are interchangeable at the CPU card level. That way, your system won't become instantly obsolete with each new design breakthrough. The major portion of your investment in memory and I/O is protected.
- Digital Group systems are complete and fully featured, so there's no need to purchase bits and pieces from different manufacturers. We have everything you need, but almost any other equipment can be easily supported, too, thanks to the universal nature of our systems.
- Our systems are specifically designed to be easy to use. With our combination of TV, keyboard, and cassette recorder, you have a system that is quick, quiet, and inexpensive. To get going merely power on, load cassette and go!
- Design shortcuts have been avoided — all CPU's run at full maximum rated speed.

**The Features**

Digital Group Systems — CPU's currently being delivered:
8080A/9080A 6800 6500 by MOS Technology

All are completely interchangeable at the CPU card level. Standard features with all systems:

- Video-based operating system
- Video/Cassette Interface Card
  - 512 character upper & lower case video interface
  - 100 character/second audio cassette interface
- CPU Card
  - 2K RAM, Direct Memory Access (DMA)
  - Vectored Interrupts
  - 256 byte 1702A bootstrap loader
  - All buffering, CPU dependencies, and housekeeping circuitry
- Input/Output Card
  - Four 8-bit parallel Input ports
  - Four 8-bit parallel Output ports
- Motherboard

Prices for standard systems as featured above, start at $425 for 8080 or 6800 and $375 for 6500.

**More**

Many options, peripherals, expansion capabilities, and accessories are already available. They include rapid computer-controlled cassette drives for mass storage, color graphics, memory, I/O, monitors, multiple power supplies, prototyping cards, and others. Software packages include Tiny BASIC Extended, games, ham radio applications, software training cassettes, system packages, and more (even biorhythm).

**Sounds neat — now what?**

Now that you know a little about who we are and what we're doing, we need to know more about you. In order for us to get more information to you, please take a few seconds and fill in our mailing list coupon. We think you'll be pleased with what you get back.

**THE DIGITAL GROUP, Inc.**
**P.O. Box 6528**
**Denver, Colorado 80206**
**(303) 861-1686**

OK, I'd like to get to know you guys better. Send me the whole package!

Name ______________________

Address ______________________

City/State/Zip ______________________

II

## NEWS FLASH — FLOPPY DISK KIT NOW AVAILABLE

*This news comes to you moments before press time, much to the dismay of our publisher.*

The information we have been waiting for is out! A floppy Disk Kit will be available through Group Purchase. This is not a cheapie! It consists of a drive, the controller boards and interconnecting cables of a design that has been in use in industry for two years — it's proven equipment.

iCOM, Inc., a leading producer of reliable Microperipherals™ is offering the equipment at approximately $1,000 in quantities of 100. Although the details are not yet known, initial word indicates the hardware, optional software and optional power supply will be available through group purchase in the near future.

This will be a "kit" only in that the purchaser must provide his own case and power supply. The critical components (drive and controller boards) are pre-assembled and tested. The optional software, which consists of operating system and hooks to add assembler, editor, etc., is the best this author has seen. It will be available for about $250.

The addition of a 1K PROM (for the disk handler) and three 8 bit latches (parallel ports) will result in one of the finest pieces of equipment available to the personal computing market.

Delivery is expected to begin approximately three weeks after Group Purchase receives the first 25 orders. The software is available for either 8080 *or* 6800 based machines.

Call our Group Purchase number (213) 425-5120 for complete details.



by building them only into the hardware (in which case we can all throw our computers in the trash and start building home brew EAM equipment) or second, put the results of our creativity, ideas and concepts where they *naturally* belong—in the public domain.

Rapidly advancing technology will solve the problem sooner than most of us realize. The chip which serves as the heart of the microcomputer will someday contain the entire logic for, possibly, several high level languages. Purchase the chip and you purchase the language.

Until that day the manufacturer who realizes that an easily utilized machine is a more desirable machine, and hence gives the interpreter or compiler to every purchaser of the machine, is going to show far greater profits than those who attempt to sell the two separately.

The final consideration is to weigh software ideology and philosophical attitudes against that which is practical. Would the software writer rather spend his hard-earned time and money pursuing copyright offenders? That same time would surely be better spent creating new software to outdate the offended (and stolen) material. He'll have a hell of a time catching *all* software thieves, hobbyist *or* otherwise.

## IN THIS ISSUE

### WAR OF THE PROCESSORS

To most hobbyists the discussion would center on the 8080 and the 6800. The question would be, "Which is the better microprocessor?" But that is only a small part of the subject. Adam Osborne takes a look at the entire picture, shares his opinions of what the future holds, and gives an explanation of the events that have taken microprocessor technology to its present point.

### SERIAL DATA COMMUNICATIONS

Terry Benson continues his series of articles with more hints on how to cut the cost of home computing by substituting software for serial interface hardware. Besides aiding the budget-burdened computer buff, Terry's series is an excellent assembly language programming tutorial.

### TELETYPE MAINTENANCE

Cliff Sparks—his gears, pawls, cams, levers and all those other funny parts and pieces are still with us. Cliff continues to help make teleprinters less a mystery, and as many readers have declared, home computing easier.

### FIREWORKS—ANCIENT AND MODERN

Watching a John Whitney computer art movie is an experience in learned appreciation. The first viewing is nice, the second is awakening, and the third is exciting. John's series on computer-generated art provides an insight into the man, and the philosophy which lies behind the talent.

AND MORE . . . Inexpensive time sharing is now available to Los Angeles SCCS members. Alden Rhodes describes the facility in "No Such Thing as Cheap Timesharing."

Sheila Clarke shares a view of computer hobbyists-turned businessmen in "Polymorphic Systems . . .," and Daryl Schatz shares his view of the IMSAII 8080 kit.

DEPARTMENTS contain several items of particular interest. One is a new game—FROG—in Games 'N Things, and the other is a hardware report on the Sphere kit by Warren Weimer in our experimental department "Up 'N' Runnin'."

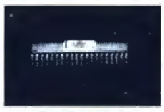
The Software Department features a text editor written in BASIC, contributed by Richard S. Edelman. Besides being a useful software tool, this program is an interesting education in the use of Basic for those who care to study the technique used.

*Editor*

## COVER STORY

National Semiconductor Corp. created this month's cover emphasizing the tremendous impact microprocessors are having today in the consumer market and industrial market. With keen competition and dynamic sales profiles, the I.C. manufacturers are vying for leadership recognition. For more information on National's broad microprocessor line contact: Director Microprocessor Marketing, National Semiconductor Corp., 2900 Semiconductor Dr., Santa Clara, Calif. 95050 (408) 732-5000.

# You can buy this microcomputer for $39.95, but...



MC 6800

We would be a bit surprised if you could do anything meaningful without additional hardware and software. **Wave Mate's Jupiter II™** isn't the kind of microcomputer kit you only stare at... when you've completed your Jupiter II just plug in your terminal and you're ready to go. That's because it goes beyond the sum of its high quality parts. It's the ultimate micro kit experience. **In performance, in documentation, in reliability.** First, consider its superb features. It has small **pluggable wire wrapped cards** easily tailored to suit your modifications. **Every IC is socketed and 100% burn-in tested.**

In fact every part including the powerful MC 6800 CPU and the 8K dynamic RAM is guaranteed for 120 days. It has the best software around, **System Monitor** and **Debug programs** (ROM). Includes powerful **text editor** and **Motorola compatible assembler.** And BASIC at no extra cost. Because we've been making microcomputer systems for over 4 years, we can offer you the broadest line of interfaces including TV terminal and dual audio cassette. Impressive. And yet the grandest feature is the experience of completing a kit that works. Guaranteed.

Wave Mate

ACT NOW AND SAVE.
SPECIAL price $1445.00 (assembled $1950.00)

Jupiter II Microcomputer Kit Includes:

- Modular plug-in power supply
- 9 module PC backplane
- CPU module
- System monitor module
- 8K dynamic RAM module
- Serial RS 232 communication interface module
- Front panel module
- Front panel
- Wire; cut, stripped, color coded
- Rack mount module cage
- Wire wrap tool
- Wire unwrap tool
- Cables, connectors, all other necessary hardware
- Software (editor, debug, assembler, BASIC)
- Assembly manuals
- Operators manuals
- Theory of operation manuals
- Annual membership in users group

NEXT MONTH WATCH FOR STUPENDOUS ANNOUNCEMENT!!

☐ Send Jupiter II micro kit.
☐ Kit ☐ Assembled
☐ Send details on kit-a-month
☐ I'm not convinced; send me 101 reasons why.
☐ Please send free Jupiter II System catalogue.

☐ Check enclosed for $ ______________
Include $10.00 for postage and handling. California residents add 6% tax Delivery 60 days ARO

Name ______________
Address ______________
City/State/Zip ______________

WAVE MATE
1015 West 190th Street Gardena, California 90248
Telephone (213) 329-8941

Wave Mate

WARRANTY
120 days on all parts, materials and workmanship.

CIRCLE NO. 3 ON INQUIRY CARD

# UPdate

MAY MEETING—Saturday, May 22, 1976
10:00 a.m. to 4:30 p.m.
The Samuel Goldwyn Theater
8949 Wilshire Blvd., Beverly Hills

COMPUTER GRAPHICS AND MOTION PICTURES is the theme of the special event offered to the membership by the combined efforts of the SCCS, The Academy of Motion Picture Arts & Sciences, The Society of Motion Picture and Television Engineers, and ASIFA Hollywood (The International Animated Film Society). The program will include John Whitney and his film, "Arabesque," Ivan Sutherland with his film, "Sketchpad," and Glen Fleck showing two films, "Computer Glossary" and "Computer Perspective."

No one will want to miss this one, but advance reservations are required. To make your reservation, a $2 donation per person must be made out to "Computer Society—Graphics," and mailed to Box 54751, Los Angeles 90054. For additional meeting information, contact Louis G. Fields at (213) 472-0388.

## MARCH MEETING

The numbers in attendance for the March meeting seemed to be unimpaired by the late delivery of INTERFACE (we're sorry). The two guest speakers held the audience fast; Bob Baskin, V.P. of iCOM thoroughly discussed the pros and cons of floppy disk. For those of us who did not take accurate notes, Bob promises to reproduce his remarks for reprint in a future issue of INTERFACE. Ted Nelson, author of COMPUTER LIB, extemporaneously titillated and amused the membership with his futuristic visions, illustrating that the somewhat cynically viewed "present" will continue to become better, easier and more inexpensive for the computer enthusiast.

*FULL HOUSE . . . Though the March meeting of the SCCS was not publicized, over 500 were in attendance.*

*Photo by Sheila Clarke*

## ELECTION NOMINATIONS

Elections for officers and directors are now being planned for the second year of the SCCS. Members are being asked to consider whether or not they would like to become a guiding force in the Society, or if they strongly feel someone else should be considered for nomination. Interested members should contact Chairman Marv Perlman for information and candidacy application by writing 11000 Dempsey Ave., Granada Hills, Calif. 91344.

*Bob Baskin discusses floppy disk in tutorial address at SCCS March meeting.*

*Photo by Sheila Clarke*

*Ted Nelson, author of "Computer Lib," guest speaker at SCCS March meeting.*

*Photo by Sheila Clarke*

*SCCS members and exhibitors discuss latest goodies at March meeting.*

*Photo by Sheila Clarke*

## SCCS CHAPTER NEWS

SANTA MONICA BAY CHAPTER will meet at the Venice Pavillion, at Windward and Ocean Front Walk (1531 Ocean Front Walk) on May 11th. The June meeting is scheduled for June 8th, both at 7:00 p.m. For more information, contact Larry Press at (213) 399-2083.

SAN FERNANDO VALLEY CHAPTER meets the first Wednesday of the month. This month it's May 5th, 7:15 p.m. at the Harvard School Cafeteria. For more information write John Scott at 2435 N. Lamer St., Burbank 91504, or call 849-4094.

## THE SAN GABRIEL VALLEY CHAPTER: AN EXPERIMENT IN NEAR-ANARCHY

The day Dan Erickson joined SCCS was the day he began organizing the San Gabriel Valley Chapter. Hearing of the San Fernando Valley Chapter, he asked President Ward Spaniol if a chapter had formed in the San Gabriel Valley and was "volunteered" chairman of the organizing committee. Dan could have declined, we suppose, insisting that his two-hour's experience in the organization was insufficient, or time too limited. But, it was apparent, even at that January meeting, that chapters would be necessary for closer contact. Discovering which of the 450 attendees lived close to him and establishing contacts would be difficult without some formal structure.

Following Ward's lead, Dan formed an organizing committee from the first dozen people he'd met whom he could strong-arm into helping. This included some people who were not members of SCCS at that time. The committee met informally at Dan's house several times, discussing concepts of chapter structure and activity, and taking care of such details as finding a meeting place, publicizing the initial chapter meeting, and signing up new chapter members. They decided early on that they would introduce minimum structure and cost into the plans and would let the full chapter membership add structure and/or cost as they saw fit. In addition, observing that most of the SCCS officers had no time to work on their computer hobbies, it was resolved to vest as little power and responsibility in chapter officers as possible. The chapter would be highly dependent on volunteers. If participation waned, the chapter would die. Given this decision in favor of near-anarchy, members are not sure that they should have had an organizing committee at all. Perhaps time would have been better spent discussing technical problems and advancing hobbies. If the chapter should die, this is exactly what will be done, but they will at least have met more hobbyists from which to draw those with similar interests and geographic proximity.

The first full meeting on March 25th enjoyed an attendance of 41. There were no formal elections, there are no dues, no refreshments, and no speakers. There *are* many interest groups within the chapter.

A meeting has been scheduled for May 27th in the library of Flintridge Preparatory School in La Canada (4343 Crown Avenue, at 7:00 p.m.). An April meeting has not been scheduled. Will the chapter survive? Will it become more highly organized? Only time and future issues of *Interface* will tell.

To get in touch with the San Gabriel Valley Chapter, write to: SCCS-SGV Chapter, P.O. Box 9459, North Hollywood, CA 91609.

If you wish a reply, send a self-addressed, stamped envelope. Remember, we have no funds.

## A CHAPTER IN CORAL GABLES?

Thomas M. Turner suggests a South Florida Chapter. Members in the area interested in cooperating in the effort are invited to contact Tom by writing 1450 Coruna Ave., Coral Gables, Florida 33156.

## OTHER CLUBS

THE TRIAD AMATEUR COMPUTER SOCIETY meets monthly in the Greensboro-Winston Salem area. Contact Doug Drye (919) 373-0040 in Greensboro, N.C., or Andy Pitts (919) 765-1277 in Winston Salem.

MICROPROCESSOR USERS GROUP is now forming in Phoenix. The group is so new that interested hobbyists may still have their say regarding how, when and where meetings and activities will form. Send your background, interests and preferences to either Harry C. Stanton, 3608 W. Davidson Ln, Phoenix 85021, or John Kromer, 1419 E. Hatcher Road, Phoenix 85020.

DENVER AMATEUR COMPUTER SOCIETY. This group seems to have its activities "up and running." They have a fine looking monthly newsletter, and regularly scheduled meetings and events. Contact them by writing P.O. Box 6338, Denver, Colorado 80206.

COMPUTER AND ELECTRONIC HOBBYISTS' ASSOCIATION OF AMERICA. The CEHAA, based in Flushing, New York, announces their intention to support an organization oriented to non-commercial aspects of computers and/or electronics. They will maintain an up-to-date program bank for use by all members which will be stored on most subjects in most computer languages. Members will receive a monthly magazine to be published by the club. For more info, write CEHAA, P.O. Box 13, Flushing, New York 11352.

CLEVELAND DIGITAL GROUP SPONSORS FESTIVAL. The COMPFEST will be sponsored by the Midwest Computer Alliance of Computer Clubs (Cleveland, Detroit, Pittsburgh, Columbus) in Cleveland on June 11, 12, and 13. For more information on the Cleveland Digital Group and COMPFEST, contact Gary Coleman, 14058 Superior Ave., #8, Cleveland 44116.

## EDUCATION UPDATE

POWER SUPPLIES—Fred Schultz, a consultant with 20 years experience, will offer a class on power supply specification and design. Discover the answers to questions like, "What sort of power supply do I need and how can I build it?" Class will be held

on Saturday afternoons for three sessions, beginning mid-May at Santa Monica College, 20th and Pico in Santa Monica. Prerequisites require that you have a basic knowledge of DC circuits (Ohm's Law). Beginners can satisfy this by reading "Basic DC Circuits" by Swan and Palmer, available at Radio Shack Stores. Tuition is $20, and scholarships are available . . . just ask. Contact Larry Press at 128 Park Place, Venice 90291, and make your check out to SCCS. Your check won't be cashed until class is confirmed.

BASIC PROGRAMMING CLASS conducted by Irv Naiman, will be held at the Harvard School in North Hollywood. Eight Monday night sessions are planned. Once again, contact Larry Press for details.

CASSETTE INTERFACE—Space is still open in the design and construction class. Indicate your interest by writing Larry Press.

### SCCS EDUCATION PROGRAM

The purpose is to extend to all SCCS members the opportunity to learn more about hardware, software, and other aspects of home computing. SCCS invites your suggestions for classes. Those who wish to teach a class, and those who would like to attend a special class may contact Larry Press, 128 Park Place, Venice, Calif. 90291.

If you wish to take a class that has been offered, contact Larry or the class instructor. Checks for tuition should be made to SCCS. Proceeds from tuition are divided equally between the instructor and SCCS.

### SOUTHWESTERN COMPUTER SHOW

A one day event, sponsored by DeNardi Enterprises, is planned in Dallas at the North Park Inn, from 1 to 7 p.m., on May 27th. Major computer manufacturers will represent their contributions to the industry. Invitations and information may be obtained by writing NDN, 95 Main St., Los Altos, CA 94022. Phone (415) 941-8440.

### SCCS IN ALBUQUERQUE

About 50 SCCS members traveled en masse to the MITS Convention the weekend of March 27th to share enthusiasms with home computerists from all over the country. The air was electric with the genius and near-genius of innovators and futuristic visionaries who don't bother to try to comprehend why *everyone* isn't doing it. They're busy creating their own marvels . . . these are the pioneers of home computing conveniences that the rest of us will call "applicances" 20 years from now.

The SCCS hospitality suite displayed Gene Murrow's ALTAIR, and a videotape display of John Whitney describing computer graphics in motion.

Other rooms were crowded with contest entries and displays using the ALTAIR in various applications. Everyone rubbed elbows and chatted with people they'd read about, corresponded with, or didn't even know existed until that weekend. Perhaps the best news to come out of the MITS Convention, is that no matter where you live with your hobby in home computing, you're definitely not alone. You probably found someone to share it with who lives in your city. We may all look forward to a similar event somewhere in the country soon.

### PRIZES AWARDED

Mr. Don Alexander of Columbus, Ohio was named Grand Prize Winner in the MITS World Altair Computer Convention Demonstration Contest with his computer-controlled amateur radio Teletype station. The home-built system consisted of an Altair 8800 with 8K of memory, an ASCII keyboard, a video display, Baudot Teletype and standard transmitter and receiver. In addition to building the hardware, Mr. Alexander developed his own software and wrote the assembler and editor for the system. The program he demonstrated at the Convention was written for receiving and transmitting messages in a radio Teletype contest. The Altair 8800 kept track of most of the radio Teletype contest "housekeeping," such as: ASCII/Buadot translation, cross-checking calls for duplication, sending the time and message number of a transmission along with lines of text that are generated by command from the keyboard. After every exchange, a log entry was printed on the Teletype, keeping a hard copy record of all exchanges. A complete Altair Floppy Disk system was awarded to Mr. Alexander for his winning entry.

A tie for second place resulted in MITS awarding two Altair 8800B's: one to Randy Miller of Tempe, Arizona for his remarkable computer chess demonstration; and one to Wirt and Valerie Atmar of Las Cruces, NM for their speech synthesizer.

Third prize, an Altair 16K Static Memory Card went to Danny Kleinman, Steve Grumette and Mike Gilbert of Los Angeles for their popular backgammon game, written in Altair BASIC and played on a Cromemco TV Dazzler.

The winners were announced on Sunday, March 28, 1976 at the Altair Awards Banquet, the event which highlighted the Convention's activities. The Grand Prize winners in MITS' yearly Software Contest were also named at the banquet; Mr. James Gerow of Houston, Texas won first place for the Best Program, and Mr. Lee M. Eastburn of Langdon, ND took the top prize in the subroutine category.

*A friend pitches in to get Gene Murrow's Altair assembled for display.*

*See? We knew we could do it!*

*WHAT ARE ALL THOSE PEOPLE LOOKING AT? John Whitney's videotape of computer graphics (which doesn't photograph well from the VT screen).*

*People crowded in to see what others had done with their ALTAIRS.*

*Art Childs, David Bunnell and Ward Spaniol discuss convention activities.*

*Convening with friends for dinner before evening events, Pearce Young, Leda Alpert and John Craig (73 Magazine I/O Editor).*

*(l to r) Randy Miller (Second Prize winner), Danny Kleinman, SCCS member Steve Grumette and Mike Gilbert (Third Prize winners).*
*photograph by Robert Prati*

*Lou Fields, Vice President of Southern California Computer Society, presents trophy to Carl Helmers of Byte, in recognition of Byte's contribution to the home computer field. Altair Awards Banquet.*
*photograph by Robert Prati*

# APPLICATIONS EXCHANGE

**COORDINATORS**
BIOFEEDBACK: Larry Press, 128 Park Place, Venice, Ca. 90291 (213) 399-2083
BIORYTHMS: Art Childs, 335 N. Adams, #210, Glendale, Ca. 91206 (213) 243-5179
GAMES: George Tate, 3544 Dahlia Ave., Los Angeles, Ca. 90026 (213) 663-2604
MITS BASIC: Jon Walden, 11557 Sunshine Terrace, Studio City, Ca. 91604 (213) 769-6569

The people listed above have volunteered to be informal coordinators for the areas shown. Would you be willing to coordinate requests for information, programs, or literature in an area that interests you? If so, let me know.

### COMPUTER SCULPTORING

I recently visited the studio of sculptor Barry Gott. A sculptor's studio might seem an unlikely place to run into microprocessor applications. Barry however, is not your run-of-the-mill sculptor. He uses plastics, light, coat hangers, lasers, tin foil, LEDs, water holographs, and more unlikely miscellaneous paraphernalia as his media. Many of his works incorporate digital electronics to control the sequencing of visual (light) events. Until now the control has been fairly simple and fixed (perhaps a 555 timer and a counter controlling a data selector), but Barry is now experimenting with microprocessor controlled sculpture! Why not let a piece of sculpture interact with the observer, change as a function of the number of people looking at it, change in response to the observer's heart rate? Why not let people alter (change program parameters of) the sculptural pieces they are "observing?" Why not?

If you are interested in computer applications in sculpture, Barry would be a good guy to get to know. He has been following his singular path for about six years and, in addition to unearthing his personal symbolic "language" has picked up a broad range of technical skills—not the least of which is a knack for scoring incredible bargains on surplus electronic components. Barry lives and works in Hermosa Beach, California and phone messages for him may be left at (213) 376-6015.

Now that you know what sorts of things Barry Gott is into, how about letting us all know what sorts of applications you are interested in and are working on. Drop me a line describing your applications so that we can pass them along through the "Applications Exchange."

### TUTORIALS

The last two issues have discussed the need for tutorial material for SCCS members who are new to hardware experimentation. The April "Applications Exchange" suggested some tutorial reading for the beginner as well as some experiments with basic DC circuits. If you have done the reading and tried a few experiments, you are probably feeling the need to play with a few of those mysterious ICs which you have read about.

The first experiment you should run with ICs is to break one open with a hammer and see what's inside. Once you have done that, you might try a little electronic experimentation for which you will need a bit of equipment: a power supply, a voltmeter, a means of conveniently connecting leads to your ICs.

### BUILD YOUR OWN POWER SUPPLY

If you have done some DC experiments, you must already have a voltmeter, but what about a power supply? I built a "one amp five volt" power supply from a Radio Shack kit (the actual performance of the power supply is shown in the following table).

| *CURRENT (Amps)* | *FLUCTUATION (From 5.4 volts)* | *FLUCTUATION (Percentage)* |
|---|---|---|
| 0.19 | 0.0 | 0.0 |
| 0.25 | 0.4 | 7.4 |
| 0.26 | 0.5 | 9.3 |
| 0.50 | 0.7 | 13.0 |
| 0.83 | 1.0 | 18.5 |
| 1.00 | 1.1 | 20.4 |

*The kit I assembled was Radio Shack's 1 amp, 5 volt power supply. The finished supply puts out 5.4 volts with no appreciable "ripple" (variation in the output voltage through time) up to a load of about .19 amps. When more current is drawn, the voltage fluctuations are as illustrated.*

The power supply will be useful for real projects as well as beginning experiments. Building the power supply took about 30 minutes, but mounting it in a neat aluminum case took three hours—that was a valuable lesson. If you build one of these kits, I would recommend that you read "How to Design Your Own Power Supplies" by Jim Huffman in the June, 1975 issue of POPULAR ELECTRONICS. The article presents an understandable overview, even if the beginner gets lost in the details. With this article for background, you will be able to understand your kit rather than just build it by rote.

Once you have a source of power, you still need a means of conveniently making connections to your ICs. I picked up a couple of IC sockets and little one-IC printed circuit boards from Radio Shack. For those who are interested in something a little more extensive, a number of companies make "breadboards" which enable you to build temporary circuits of 4 or more ICs. These are somewhat easier to use than the individual sockets which I used, but the minimum investment seems to be about $8.50 (for a breadboard which will hold 4 ICs).

At this point you should be ready to try out a few ICs—to see if those truth tables tell the truth. William Browning's article, "The Best Logic Yet" in the August issue (1975) of "73" Magazine is a

Branch to pg. 60

# Letters to the Editor

**The editor of INTERFACE Magazine wants to know how YOU feel, what YOU think, and respond to your suggestions and requests. Write Editor, INTERFACE Magazine, 335 N. Adams St., Suite 210, Glendale, Calif. 91206.**

Dear Friends:

Do you have any other SCCS members who live in Marin County (949XX zip codes)? If so, I am interested in starting a group up here (maybe a chapter of SCCS).

Dave Melnick
533 Miller Creek Rd.
San Rafael, Calif. 94903

---

Dear Editor:

There are several faculty members here at Stephen F. Austin State University (computer professionals and hobbyists) who enjoy your magazine and may wish to become members of SCCS. Should we want to form a "local chapter" of SCCS, are there certain formal steps we must follow, or may we simply refer to ourselves henceforth as the Nacogdoches, Texas Chapter of SCCS?

Camille C. Price
Dept. of Computer Science
Box 6167, SFA Station
Nacogdoches, Tex.

*Dear Ms. Price:*

*That's the first step. Let us know when you are organized.*

*Editor*

---

Dear Don Tarbell:

I have been going over your article, "More Basic Than Basic," in the February '76, page 34, issue of INTERFACE, and I don't understand the instruction on line 400 of the program you had listed. Could you define this function and its limits for me. I don't have an 8080 but am studying how to program them. I have access to a Honeywell 6000 and Sigma 9 systems. Thanks.

Roger W. Brown
Utica, New York

---

Dear Roger:

Although there are other misprints, line 400 is not one of them. The PEEK (N) function returns the contents of memory address N, in decimal. This is unique to Altair BASIC, as far as I know. The lines (corrected) which actually *were* misprinted are listed below:

```
600 IF X < 32 OR X > 95 THEN Y = 32
700 IF X > 159 AND X < 224 THEN Y = X - 128
1025 DATA ADC,B, ADC C, ADC D, ADC E, ADC H,
     ADC L, ADC M, ADC A
```

Don Tarbell

---

Dear Mr. Childs:

A brief comment concerning the Editor's Note accompanying the article, "A 4K Memory Check," commencing on page 31 of the January, 1976, INTERFACE.

". . . unspoiled mind . . ." is not the way we see it. When someone chooses to do what is easy for himself at the expense of making it harder for the vast majority, we think the proper adjective is "spoiled." Not only was Jon not biased by hardware engineering training, he was not biased by any consideration for all the readers out here who have managed to become accustomed to the defacto standard.

Instead of condoning Jon's inconsideration, the staff should have converted the program to the accepted standard format as a service to readers.

Tim Barry
Software Design Engineer
Mountain View, Calif.

---

Dear Sir:

I have just finished reading the December issue of INTERFACE. I enjoyed it so much that I am sending you my check for $10.

I would like to make several comments about the magazine. Regarding Lloyd Rice's "Toward the Design of a Micro-Operating System," I applaud his efforts. I agree with most of his article, especially the need for modular construction. However, I do not think that an operating system can be designed without regard for the nature of the mass storage device. Its speed and access capabilities to a great extent determine the form the operating system must take. Having designed, programmed, and debugged a very simple cassette operating system for an H-P 2114, I am well aware of the man months involved in such a project and I would like to suggest that the Society assign to one of its committees the task of outlining in at least rough fashion the structure of several types of operating systems, the basic modules involved, and methods of parameter passing. If this were done, it might enable people to help each other instead of each person being forced to invest those hours all by himself. Also, I would like to see a feature added to your magazine. That feature is a survey of equipment. A typical article might, for instance, survey the new TI microprocessor, listing instructions, cycle times, basic structure, sources, prices, and an address for more information.

Lastly, I wonder if anyone else is interested in a group purchase of all or part of Monolithic Memories new Series 300 minicomputer.

John Franklin Jones, Jr.
569 Park Meadow Drive
San Jose, Calif. 95129

# Can anyone beat the Altair™ System?


# We doubt it.

**When it comes to microcomputers,** Altair from MITS is the leader in the field.

The Altair 8800 is now backed by a complete selection of plug-in compatible boards. Included are a variety of the most advanced memory and interface boards, PROM board, vector interrupt, real time clock, and prototype board.

Altair 8800 peripherals include a revolutionary, low-cost floppy disk system, Teletype,™ line printer, and soon-to-be-announced CRT terminal.

Software for the Altair 8800 includes an assembler, text editor, monitor, debug, BASIC, Extended BASIC, and a Disk Operating System. And this software is **not just icing on the cake**—it has received industry wide acclaim for its efficiency and revolutionary features.

**But MITS hasn't stopped with the Altair 8800.** There is also the Altair 680—complete with memory and selectable interface—built around the new 6800 microprocessor chip. And soon-to-be-announced are the Altair 8800a and the Altair 8800b.

**MITS doesn't stop with just supplying hardware and software, either.** Every Altair owner is automatically a member of the Altair Users Group through which he has access to the substantial Altair software library. Every Altair owner is informed of up-to-date developments via a free subscription to **Computer Notes.** Every Altair owner is assured that he is dealing with a company that stands firmly behind its products.

After all, we didn't become the leader by messing around. Shouldn't you send for more information or visit one of our Altair dealers?

**Altair Coupon**

Please send me the following information:

- ☐ Your latest catalog and price list
- ☐ Software information package
- ☐ Please include a list of your dealers

NAME____________________

ADDRESS____________________

CITY________________ STATE & ZIP________

mits
2450 Alamo S.E. Albuquerque, N.M. 87106



CIRCLE NO. 5 ON INQUIRY CARD

# THE WAR OF THE PROCESSORS

Most microprocessors were designed as shots in the dark. Now the manufacturers are madly scrambling to create markets for their products. That is the scenario for "The War Of The Processors."

To understand what is happening, we must begin by identifying the microprocessors on the market today. In the short space of this article I cannot hope to describe a dozen products in detail, rather I will categorize microprocessors by type—ranging from the most minicomputer-like, to the most digital logic oriented—then a discussion of the markets forming around each type of product will make more sense.

The present crop of microprocessors started with two parallel products: the Intel 8008 and the National Semiconductor IMP8 and IMP16.

Both of these products made their appearance in 1972. The Intel 8008 was developed by Intel under contract to Datapoint Corporation of San Antonio, Texas. Datapoint Corporation provided Intel with the 8008 specification and funded the development of prototype chips. The prototype chips developed by Intel were too slow to meet specification, so Datapoint Corporation declined to buy in volume; and Intel was left with a product that they were free to use in any way. Intel elected to sell the 8008 as an 8-bit microprocessor and it sold very well.

Now the key to the Intel 8008 was that Datapoint Corporation intentionally specified a product that was simple enough to implement on a single chip. In other words, they kept instruction sets simple and addressing modes primitive to meet the limitations of single chip LSI technology, as it existed at that time.

By way of contrast, National Semiconductor elected to build a microprocessor that would provide minicomputer-like instruction sets and addressing modes, irrespective of the number of LSI chips that would be required. Thus, at the time Intel was coming out with the 8008 on a single chip, National Semiconductor was coming out with the IMP-8 on five chips, or the IMP-16 on seven chips. The IMP product line used the "bit slice" philosophy. A CPU was implemented by stacking 4-bit wide Registers Arithmetic and Logic Unit (RALU) chips; thus the IMP-8, being an 8-bit microprocessor, required two RALU chips; the IMP-16, being a 16-bit microprocessor, required four RALU chips. In addition, each microprocessor needed a Control Unit chip and a couple of interface chips.

What was the tradeoff? Intel offered the single chip 8008, a pretty hokey device when compared to minicomputers, but being on a single chip, the 8008 was very cheap. National Semiconductor offered microprocessors that looked like simple minicomputers; in fact there is a heavy Nova minicomputer flavor in the IMP microprocessor architecture; but being implemented on five or more chips, the IMP products were much more expensive than the Intel 8008. Thus, the first war of the microprocessors was between the cheap and crude Intel 8008, versus the expensive and far more minicomputer-like IMP-16.

Even at this early stage, it became apparent that there was a bigger market for cheap devices with limited performance and Intel 8008 sales completely overwhelmed National Semiconductor.

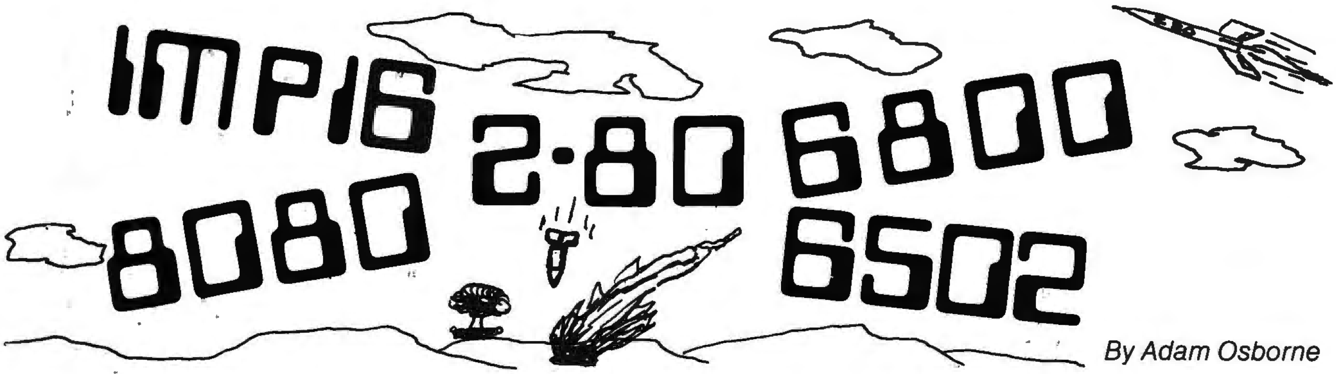
Now it was time for a second wave of microprocessors. This second wave consisted of these four product types:

1. Intel's enhancement of the 8008 and National's IMP enhancement.
2. Other manufacturers' enhancements of the Intel 8008 and the IMP-16.
3. Completely new microprocessors that do not copy or enhance anything.
4. Microprocessors that duplicate the instruction sets and addressing modes of existing minicomputers.

Let us look at these categories one-by-one.

Intel's second offering was the 8080, which is a rather obvious enhancement of the 8008. Having learned how to pack a lot more logic onto a single chip, Intel removed the most obvious faults of the 8008. The 8080 has direct addressing, where the 8008 had implied addressing only; the 8080 has a stack, a limited number of 16-bit instructions and it has separate address and data busses. Thus, the 8080 can execute more powerful programs and it is easier to interface to.

National Semiconductor's second offering was PACE, a one-chip version of the IMP-16. National Semiconductor made a tactical blunder when coming out with PACE. When the two IMP microprocessors got severely beaten by Intel, National Semiconductor wrongly attributed the whole IMP problem to price, resulting from having too many chips. National ignored the fact that the Intel 8008, with its inferior performance, was adequate for most applications. Indeed, building a 16-bit single-chip microprocessor,




By Adam Osborne

where Intel was building an 8-bit, single-chip microprocessor may have seemed like a winning strategy on paper, but there were two flaws in this reasoning:

First, the more complex, 16-bit, single-chip microprocessor is slow; in many cases PACE offers no significant advantage over the Intel 8080. Second, the 16-bit, single-chip microprocessor is more expensive than the single-chip, 8-bit microprocessor—which meant that it is still going to lose all those markets for which the 8-bit microprocessor is adequate; and the 8080 had already proved the large size of these markets. History has shown that although PACE has acquired a small and dedicated following of users, the Intel 8080 has outsold it more than ten to one.

Now look at the second category of microprocessor manufacturers: those who attempted to second guess Intel and National Semiconductor at their own game. Foremost in this category is Motorola. The M6800 seems to have been a clear attempt to enhance the Intel 8008 and participate in its markets. (The Intel 8080 and the Motorola M6800 both have second source manufacturers, but I am not going to separately categorize identical microprocessors, simply based on their second source.) Both the Intel 8080 and the M6800 have spawned second generation enhancement products. MOS Technology has generated a family of microprocessors which may be looked upon as a super set of the Motorola M6800; that is, the MOS Technology microprocessors will do everything that the M6800 will do, and they will do more. Similarly, Zilog manufactures the Z80, a microprocessor that will do everything that the Intel 8080 will do, and a great deal more.

When we turn to National Semiconductor, the situation becomes somewhat more hazy. No manufacturers enhanced the IMP-16, the way Motorola enhanced the Intel 8008. The PACE market was simply too small to attract such attention. Rather, there is a range of microprocessors with minicomputer-like instruction sets and addressing modes; it is hard to say whether these microprocessors were influenced by PACE, or simply by the designers' assumption that minicomputer-like characteristics were desirable. General Instruments' 1600 microprocessor is probably the most similar to PACE. The Signetics 2650 and the Electronic Arrays' 9002 have very minicomputer-like instruction sets which may have been influenced by PACE, but are probably better categorized as group 3 products, that is, products designed on their own merits.

National Semiconductor's SC/MP may be looked upon as the descendent of the PACE; it is more accurately viewed as the grandson of the IMP-16, rather than as the son of the IMP-8. SC/MP has minicomputer-like instruction sets and addressing modes, though not as minicomputer-like as either the Signetics 2650 or the Electronic Arrays' 9002.

Let us now turn our attention to the group 3 products, that is, those that were designed on their own merits. We will list these products in a rational order, beginning with the most minicomputer-like (excluding the group 4, minicomputer copies) and ending with the microprocessor most unlike a minicomputer.

First, there are the Signetics 2650 and the Electronic Arrays' 9002, which we have already mentioned in group 2. These are both 8-bit microprocessors with sophisticated, minicomputer-like instruction sets and addressing modes.

Now comes the great divide, separating minicomputer-like microprocessors from the very digital logic oriented products. In this category you will find the Fairchild F-8, the Rockwell PPS-8 and the RCA COSMAC. These products have instruction sets and addressing modes which might leave a minicomputer programmer shaking his head in disgust or bewilderment, but they are extremely attractive to a user who wishes to replace digital logic.

Finally there is the SMS-300, which is a strange animal. This is a microprocessor built strictly for control applications. Using bipolar technology, it is lightning fast; it executes entire instructions in 300 nanoseconds, but an SMS-300 instruction is very hard to compare with any minicomputer instruction. If you try to perform data processing operations, the SMS-300 instruction looks very primitive and it takes three or four of these instructions to do what one minicomputer instruction will do. If, on the other hand, you are controlling signals and switches, one SMS-300 instruction will do as much as three or four minicomputer instructions. For its own specialized class of applications, therefore, the SMS-300 can out-perform most minicomputers.

Now for group 4 microprocessors. And these are growing the fastest. The first microprocessor to be simulated exactly was the PDP-8, offered as a single

chip CPU by Intersil and called IM6100. Data General now offers the microNOVA, a Nova minicomputer look-alike. Western Digital has built a three-chip PDP-11 CPU; offered by Digital Equipment Corporation is the LSI-11. Similarly, General Automation offers an LSI version of the SPC-16. We can safely say that in the near future just about every commercially viable minicomputer will have an LSI implementation.

In summary, we can categorize the microprocessors on the market today with an illustration such as Figure 1. In this figure microprocessors are located in ascending order of minicomputer-like characteristics, and date of commercial production.

Now as a microcomputer hobbyist, you may look at Figure 1 and immediately assume that as a microcomputer starts to look more like a minicomputer, it becomes a more desirable product. If you are planning to write a large number of programs, this logic has merit, since the more minicomputer-like microprocessors are easier and therefore quicker and cheaper to program.

An industrial microprocessor consumer may not be quite so impressed with minicomputer-like features. To make my argument easily understood, I will take the extreme case of the automobile industry. It is generally thought that by the early 1980's every automobile will include two or three microprocessors, controlling timing and instruments. General Motors, on its own, is potentially capable of buying ten million microprocessors a year. Whether they spend $50,000 creating appropriate control programs using a relatively inefficient assembly language, or $30,000 doing the same job with a more efficient assembly language is almost irrelevant. A $20,000 savings become one-fifth of 1¢ per microprocessor. The General Motors engineering staff will pay far more attention to the total number of chips required and to the cost of assembling microprocessor components into the automobile during manufacture. Even more important to General Motors will be questions of reliability and availability. Imagine the cost of having to recall one million cars to replace a batch of defective CPUs; imagine the chaos that would result if entire assembly lines had to be shut down because microprocessor manufacturers were unable to deliver products on time. The industrial microprocessor consumer's ordering priorities will be:

1. Reliability of individual devices
2. Availability of individual devices
3. Cost of individual devices

In contrast, the hobbyist selects microprocessors based on the following priorities:

1. Price of an entire computer system rather than the cost of individual devices;
2. Software support available and ease of programming;
3. Quality of support documentation.

In between General Motors at one extreme, and the individual hobbyist at the other extreme, you will find every shade and variation of microcomputer user.

We can therefore conclude that there is no universal set of winning features that will make one microprocessor overwhelm all others. When it comes to microprocessors, what is good for General Motors is decidedly not good for the computer hobbyist.

So which of the microprocessors shown in Figure 1 is likely to succeed? Let us make the very silly assumption that no new microprocessors appearing in the next few years will have a significant impact on the industry as a whole. Based on this assumption, I would make the following predictions:

1. There is going to be an enormous increase in volume of sales over the next few years. The bulk of this volume will be generated by a few customer companies, such as the automobile manufacturers, who will start buying in massive quantity. These large volume purchases will help the least minicomputer-like of the microprocessors shown in Figure 1, with the exception of the SMS-300, which is too highly specialized.
2. The most minicomputer-like of the microprocessors will enjoy a much smaller growth rate, but once established their markets are likely to be far more stable. The fact that General Motors can completely reprogram its microprocessors for one-fifth of 1¢ per automobile means that it can easily switch from one product to another. By contrast, a low volume buyer who has invested large amounts of money programming the Signetics 2650 will be locked into this product almost indefinitely.
3. The microprocessor market has not established itself sufficiently for any product, even the Intel 8080, to be considered certain of success or dominance for the foreseeable future. I believe that the microprocessor market will expand at least 10-fold in the next five years, which means that not more than 10% of the short-term market is yet active. Suppose Intel controls 50% of this market; that means it controls 5% of the potential short-term market. 5% is simply not a dominant position. That is not to say that the Intel 8080 is in imminent danger of being eclipsed by a new product; what it means is that virtually every other manufacturer still has the opportunity to gain eventual dominance.

What does this mean to the hobby market? It means that hobbyists and manufacturers supplying products for the hobbyist must look after themselves. They must adopt a parasitic attitude, whereby they will seize every opportunity to wring benefits for themselves out of developments in the microprocessor industry at large; but they must not for a minute think that any manufacturer of substance will be swayed by the needs of the hobby market; this market is likely to become an ever smaller percentage of the total as the years roll by. ◻

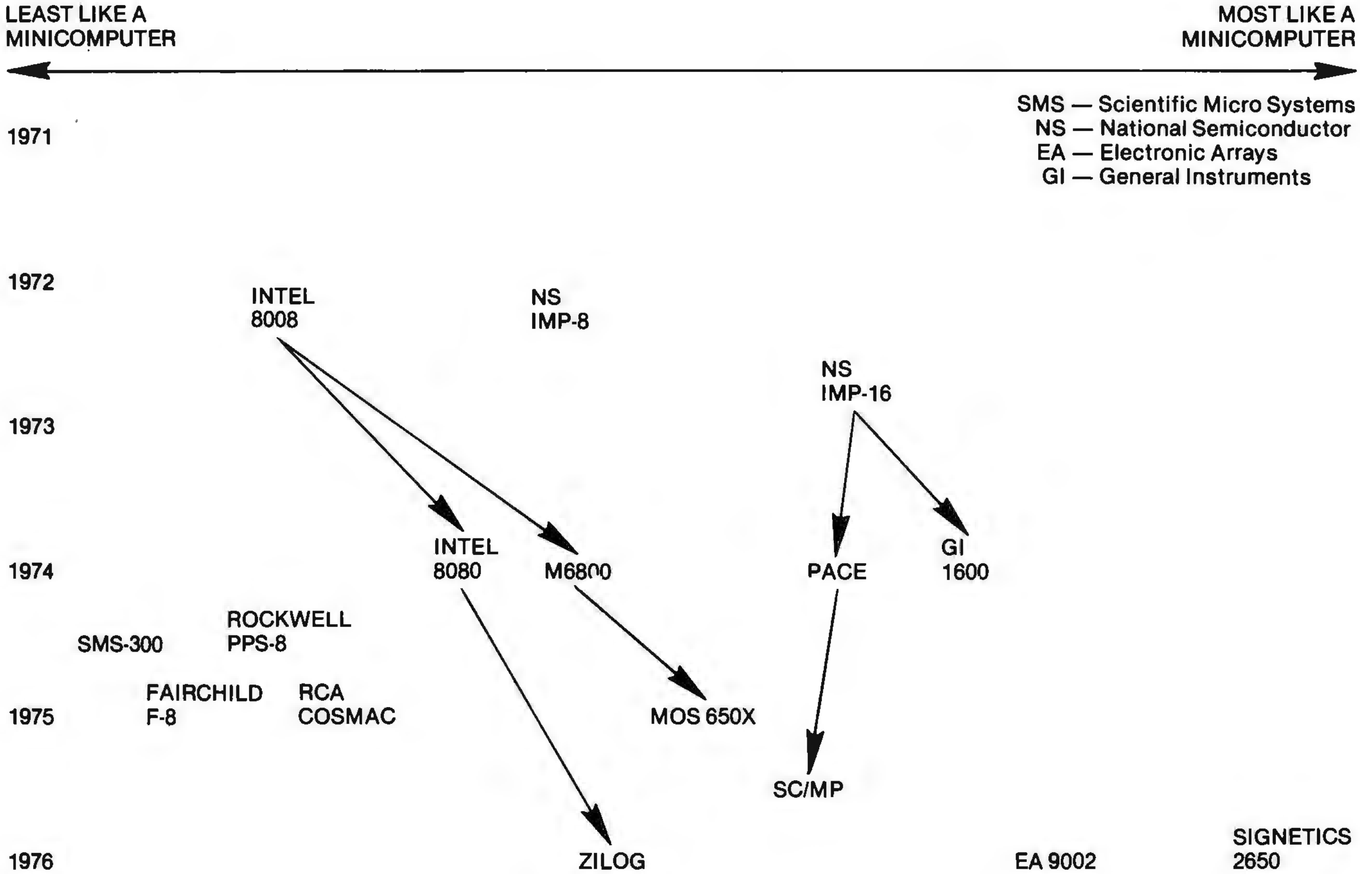
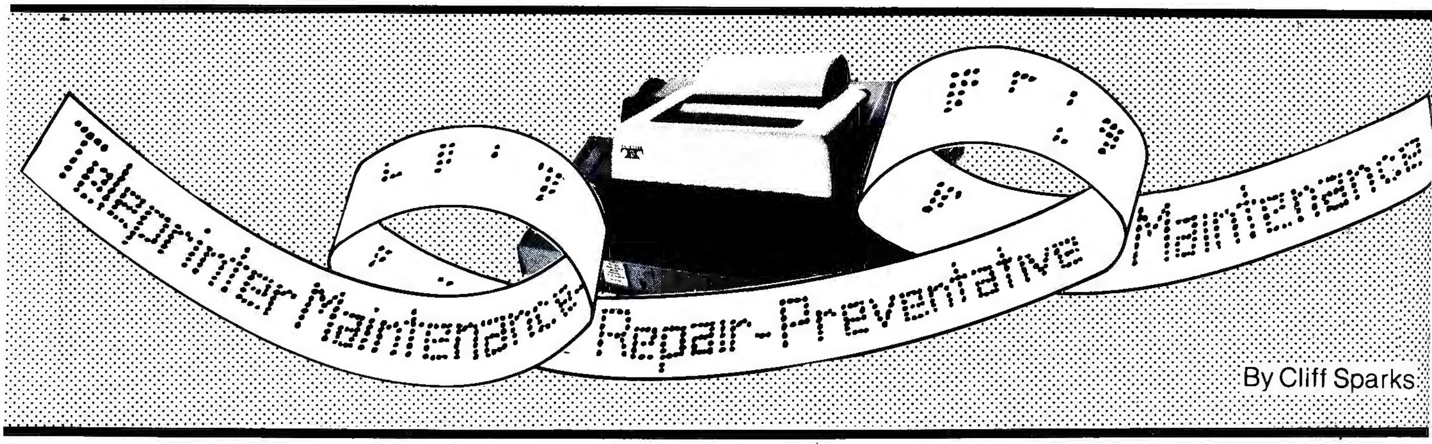



**Figure 1**
**The Evolution of Microprocessors**




This is the seventh article in a series designed to prepare the reader to perform maintenance on the model 33ASR.

The 33ASR tape punch is an 8-level device that perforates paper tape according to ASCII code. The punch is a slave that operates from commands given it by the printer codebars. There are two basic types of tape punches available. The automatic and the manual.

The manual punch is turned on or off manually. It has four pushbuttons: ON, OFF, B.SP. (Backspace) and REL. (Release).

The automatic punch can be turned on or off both manually and automatically. For manual operation these push buttons are present on the lid: ON, OFF, B.SP., REL. In automatic operation of the tape punch will turn on upon receipt of the DC2 code and turn off upon receipt of the DC4 code.

We sometimes find 33ASR sets equipped with manual/automatic tape punches. The punch, as shipped from the factory, has two clips installed in slots A-0 and A-8 which enable the punch for manual operation. Removing the clips enables the punch for automatic operation. Refer to Figure 1 for position of the A-0 and A-8 slots. Look for two small copper colored clips attached to the slotted post at the rear of the tape punch.

Figure 2 is a Functional Diagram that illustrates that the rocking action of the typing unit function rocker shaft is imparted to the tape punch by means of a sleeve which connects to a plate with a shaft (see Figure 3). A drive link, attached to the plate with a shaft, connects to a drive post which simultaneously drives the tape nudger, feed pawl,

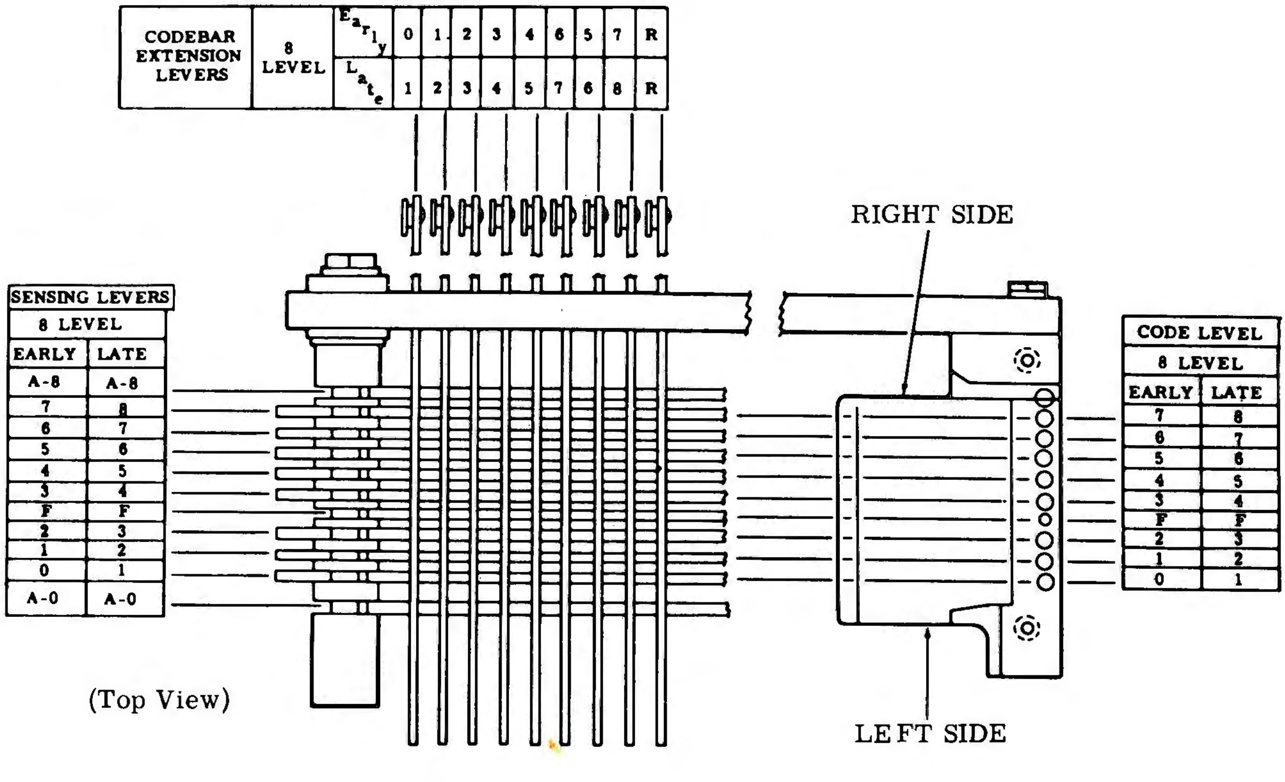



**Figure 1**
**Tape Punch Code Level Cross Reference Chart**

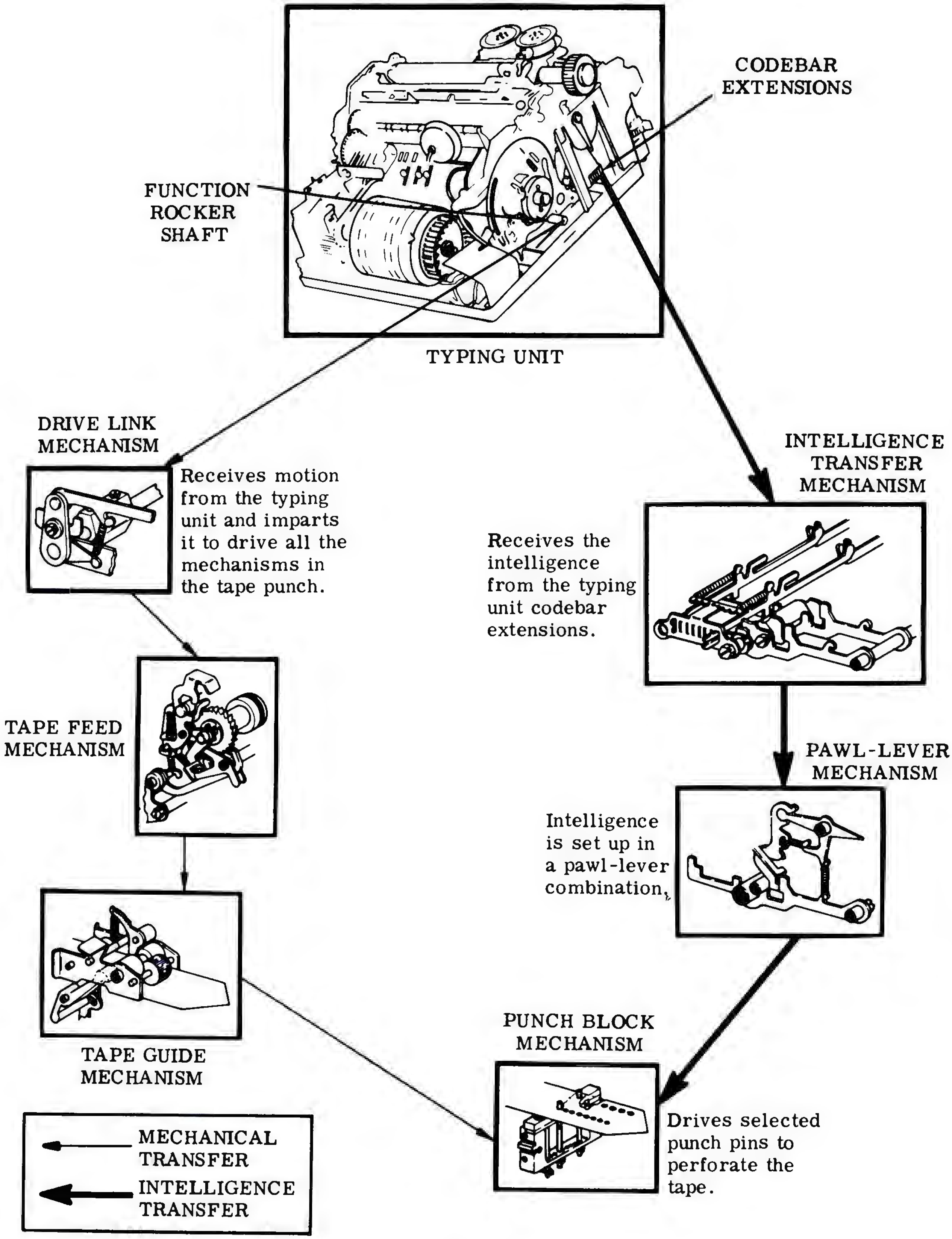



**Figure 2**
**Functional Diagram of the Tape Punch and Major Mechanisms**

and stripper bail, and supplies the downward force to pull the selected pawls by means of the sensing lever bail.

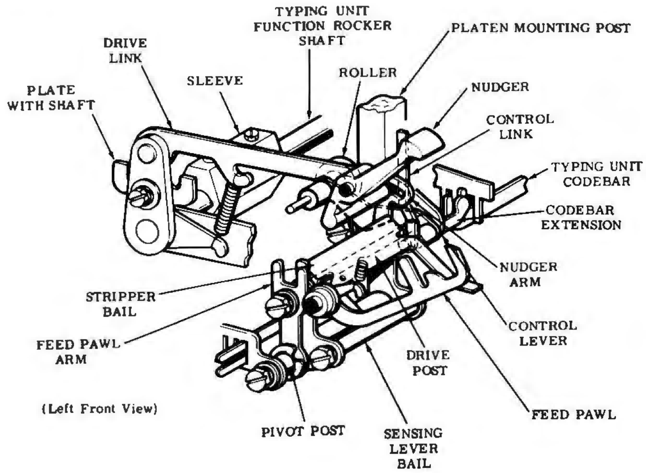



**Figure 3**
**Drive Link Mechanism and Drive Mechanism**

Figure 4 is a diagram of the Intelligence Transfer Mechanism. Each of the typing units codebars have a codebar extension. Motion is imparted to the codebar extensions by the codebars through the typing unit reset bail. A plate mounted to the tape punch side frame guides the codebar extensions. The typing unit selector blocking levers control the Mark or Space position of the codebars which, in turn, transfer this position to the codebar extensions. A blocking codebar represents a space; an unblocked codebar is a mark.

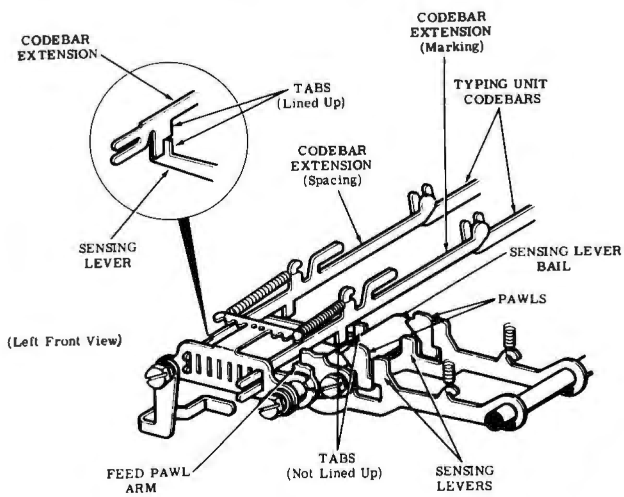



**Figure 4**
**Intelligence Transfer Mechanism**

Each codebar extension has a tab on its underside which lines up with its respective sensing lever, pawl, lever, and punch-pin combination. During the mechanism's counter clockwise travel, the sensing levers, under spring tension, move up and sense the codebar extensions. Each sensing lever, except the feed lever, has a tab on its top side which lines up with its respective codebar extension.

When the codebar extension is spacing, the tab, located on its underside, lines up with the tab on the sensing lever. The tabs engage each other, and the sensing lever is blocked from pivoting to its most clockwise position. In a marking situation the sensing lever pivots to its most clockwise position. The feed sensing lever travels to a clockwise position (it has no tabs). This moves the pawl lever, feed-punch pin combination through a latching surface on the pawl.

Each pawl and sensing lever is in its highest vertical position when the tape punch is in its off position.

A careful study of Figure 4 will reveal that the sensing levers when engaged (marking) with its associated pawl will put that lever in a selected position. As the drive mechanism (Figure 3) rotates clockwise, the feed pawl slides along the inclined surface of the adjacent ratchet tooth, drops behind it, and is cammed away from the feed wheel ratchet.

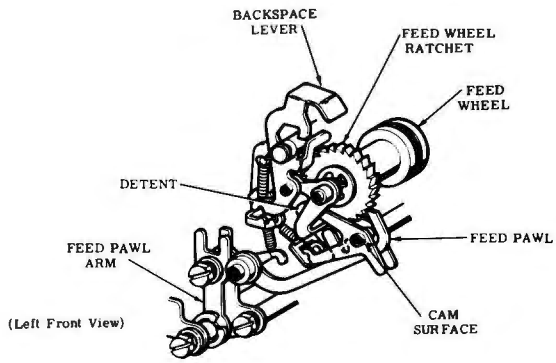



**Figure 5**
**Feed Wheel Mechanism (Tape Feed Mechanism)**

At the same time, the sensing levers rotate counter clockwise and the selected marking levers transfer their motion to the lever, and code punch pin combination. The non-selected sensing levers (spacing) receive no motion; this results in no perforation of the tape, since the code-punch pins remain in the most downward position below the tape's surface.

The tape feed mechanism operates when the stripper bail moves to the rear, the feed pawl engages a tooth on the feed wheel (Figure 5) ratchet. When the stripper bail completes its travel to the rear, the feed wheel ratchet has indexed one full tooth and the tape is advanced 0.100 inch by the feed wheel.

The tape guide mechanism (Figure 6) has a bracket, two rollers, three posts, a sleeve, and a compression spring held together by retainers. A tension spring biases the tape guide in a clockwise direction. The knurled roller settles against the knurled feed wheel. A push button located in the cover lid, when manually pushed down against a tab located on the REL. bracket, disengages the tape guide assembly from the feed wheel, thereby providing easy tape removal from the tape punch.

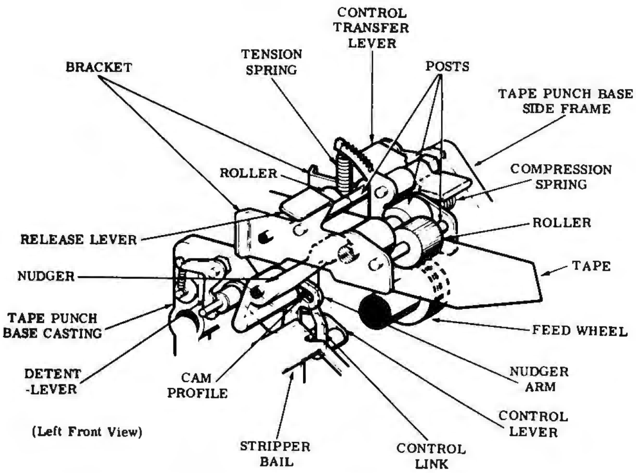



**Figure 6**
**Tape Guide Assembly (Tape Feed Mechanism)**

The punch block assembly is made up of code-punch pins, a feed-punch pin, holder, die plate, and a tape bias spring (see Figure 7).

The code-punch pin and feed-punch pins are aligned to the die plate through slots which engage levers for their respective code level. The tape bias spring always biases the tape against one edge of the holder. This results in the code hole and feed hole relation to the tape edge to be held constant.

The backspace lever (Figure 5), when depressed manually to its most downward position, backspaces the feed wheel ratchet one tooth space.

This results in the tape being backspaced one full character.

The next article will continue with more detailed information about the print carriage, and introduction to the tape transmitted. ◘

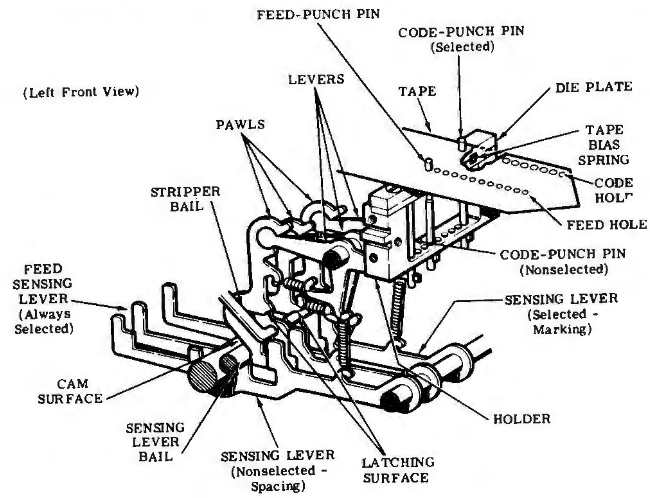
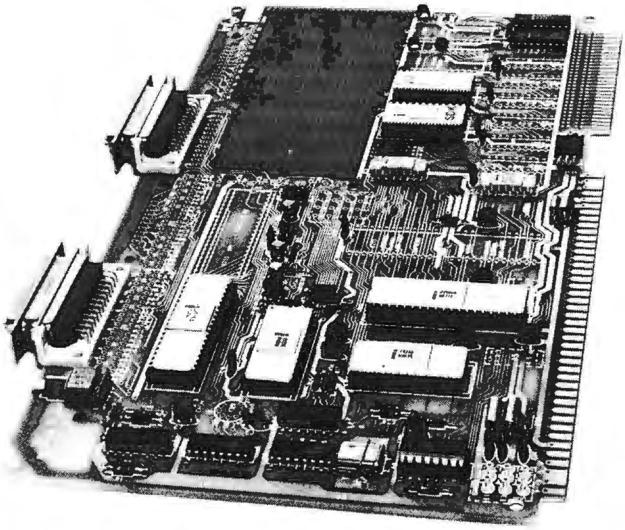



**Figure 7**
**Tape Punch Mechanism**



# Microcomputer Applications

**Serial Data Communication—Part 2**

*by Terry Benson*
*Field Applications Engineer, Intel Corp.*

If you have been following this series of microcomputer application articles, you should be familiar with various software timing techniques. For those readers who are joining us for the first time, I will provide a brief review of the topics already covered. The major concepts discussed concerned timing loops in the program to provide input and output timing for conventional *hardware* functions.

In the first article, I generated a program that could be used to output a pulse with a predictable period. The second article expanded on this concept to generate a square wave. In addition, a program was developed that could be used to measure the duration of an incoming signal. Serial data communication was then discussed and I described a program that could be used to convert an eight bit word into a serial data stream for communication with a teleprinter. Last month, all timing features were combined into a single program for data communication. The program included not only the generation and reception of a serial data stream but also a unique feature of automatic "baud rate" synchronization. This month I will illustrate the completed program and discuss the trade-offs when using a microcomputer for this type of application.

## The Complete Program

The flow chart in Figure 1 illustrates the functions that need to be performed to implement a complete serial communication program. (For more detailed flow charts refer to the two previous articles.) In the flow charts illustrated here, only the major functions that are performed are shown, but it should be sufficient for you to follow the program, especially when assisted by the program comments. (It should be understood by the first-time readers and those unfamiliar with flow charts that it is worthwhile to generate adequate flow charts prior to writing a program. The flow charts in Figure 1 are not detailed enough for program generation, but are useful for discussion purposes.)

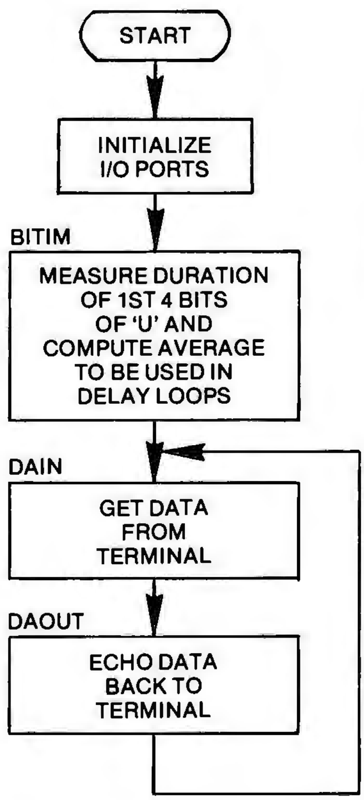



**Figure 1(a)**
**Main Program**

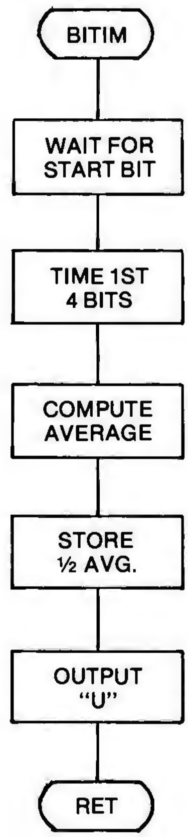



**Figure 1(b)**
**Bit Timing Routine**

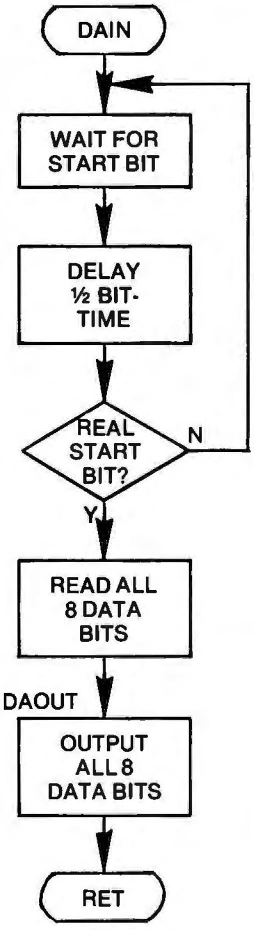



**Figure 1(c)**
**Data Input Routine**

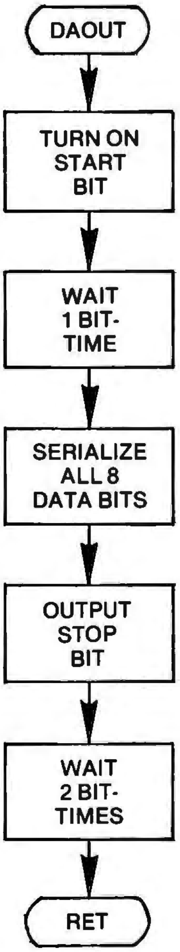



**Figure 1(d)**
**Data Output Routine**

As suggested in the previous article, some new programming methods have been incorporated into this program to stimulate the novice programmer. Note, especially, the delay subroutine implementation: last time I suggested that the ½-bit-time routine be executed twice to provide one full bit-time. The approach used in this program multiplies the ½-cycle-time value by 2 before the delay loop counter is entered. This approach facilitates the fine adjustments that are required in many types of delay applications.

The program was written to operate on an Intel 8080 and was assembled to run on the Intel MCS-80 System Design Kit (SDK-80). The listing of the program (Figure 2) contains the user generated assembly language code (source) along with the assembler generated machine language (object). The numbers in the left-hand column indicate the memory address of the instruction and will be useful in discussing and locating individual instructions. For example, the I/O set-up occurs between address 3H and 9H. (All numbers are listed in base 16—hexidecimal or, more simply, hex—hence the "H".) The main program (0EH to 11H) is used merely to test all subroutines and simply monitors the data from the terminal, then "echos" the character back to the terminal. Since the intention of these computer application articles is to provide ideas for programming techniques, individual instructions will not be discussed. If you are unfamiliar with the 8080 assembly language or the actual operations performed by any particular functions, there are several publications[1] now available that discuss the 8080 in detail. The concepts developed in this and future application articles will be general in nature and can, for the most part, be applied to any microcomputer.

Since the SDK-80 uses a crystal frequency of 18.432 MHz, one 8080 clock period is 488 nsec. However, no program changes will be required in this example even if a 500 nsec clock period is used. The reason for this is that the bit timing routine counts clock cycles and not time. Likewise, the input and output delay functions use the clock cycle counter determined in the bit timing routine (BITIM). The only function affected is the debounce routine (DBOUN), but that time, as will be seen, is not critical. In microcomputer systems where program instructions actually "time" events, the exact clock period will have to be taken into account.

## Bit Timing

The bit timing subroutine (BITIM) is the routine that determines the bit timing from the first character ("U") transmitted from the terminal. The method for determining the bit timing requires that the first four bits be measured with the average of the four values becoming the bit time used for all subsequent serial communication. It then becomes a simple matter to increment a counter for each excursion through the loop while waiting for a change on the input line. One additional requirement, not included in the original flow chart but shown to be necessary, is the debounce function. This is required primarily for mechanical devices such as the teletype but might be useful with any terminal. The reason is that the program is waiting for a second transition to define the end of a bit but mechanical "bounce" tends to create unwanted transitions that will give erroneous values of bit time unless these transitions are bypassed. Figure 3 illustrates the potential transitions which are ignored by the 200 μsec delay in this program. This value can be adjusted as required but any major change will require compensation in the preset of the bit timing counter (22H). In fact, the debounce function will probably have to be removed completely for transmission rates above 2400 baud since bit timing begins to approach the debounce time.

In the bit timing loop (25H to 2BH), 38 clock cycles are executed. This means that the value in the cycle counter is actually indicating the number of 38-clock-cycle periods. This suggests that the delay subroutine, which will later use this value, should also have a 38-clock-cycle loop—and, indeed, it does (92H to 99H). This is no accident, as you may have noticed! It is for this very reason that the extra instructions at 26H (MOV A,A) and 97H (NOP) were added. The timing balance for these two routines is essential for accurate program performance.

```
ISIS 8080 MACRØ ASSEMBLER, V1.0                    PAGE 1


                ;
                ;THIS PRØGRAM DEMØNSTRATES AND TESTS THE USE ØF A
                ;MICRØCØMPUTER FØR SERIAL DATA CØMMUNICATIØN.
                ;
                ;THE PRØGRAM IS ASSEMBLED TØ ØPERATE ØN AN INTEL
                ;MCS-80 SYSTEM DESIGN KIT (SDK-80).
                ;
                ;
                ;CØNSTANTS FØR I/Ø PØRT IDENTIFICATIØN
                ;
00F5            PTIN    EQU     0F5H    ;INPUT PØRT "B" ØN 8255
00F4            PTØUT   EQU     0F4H    ;ØUTPUT PØRT "A" ØN 8255
00F7            PTMØD   EQU     0F7H    ;MØDE SET-UP ØN 8255
                ;
1300            ØRG 1300H               ;BEGINNING ØF RAM SPACE
1300            DLYCT:  DS 2            ;2 RAM LØCATIØNS FØR CØUNTER
                ;
0000            ØRG 0
0000 313013     START:  LXI SP,1330H    ;STACK PØINTER INITIALIZATIØN
                ;
                ;SET UP I/O
                ;
0003 3E82               MVI A,10000010B ;8255 CØNTRØL WØRD
0005 D3F7               ØUT PTMØD       ;MØDE SET-UP
0007 3E01               MVI A,1         ;TØ "SILENCE" TERMINAL
0009 D3F4               ØUT PTØUT       ;ØUTPUT STØP BIT
                ;
                ;THE "BIT TIME" SUBRØUTINE READS THE FIRST INPUT FRØM
                ;THE TERMINAL AND DETERMINES THE BIT TIMING.
                ;
000B CD1400             CALL BITIM      ;TIME THE BITS FRØM A 'U'
                ;
                ;THE FØLLØWING WØULD BE THE ACTUAL PRØGRAM.
                ;HERE IT IS USED MERELY TØ TEST THE RØUTINES.
                ;
000E CD5600     KEYBD:  CALL DAIN       ;GET DATA FRØM TERMINAL
                ;
                ;AS SØØN AS DATA IS READ AND ECHØED, THE
                ;PRØGRAM WILL BE EXECUTED AGAIN.
                ;
0011 C30E00             JMP KEYBD
                ;
                ;THIS IS THE BIT TIMING RØUTINE
                ;
0014 0604       BITIM:  MVI B,4         ;TØ CØUNT FIRST 4 BITS
0016 110001             LXI D,0100H     ;SET REG D = 1 AND REG E = 0
0019 DBF5       WAIT:   IN PTIN         ;WAIT FØR START BIT
001B A2                 ANA D           ;MASK ALL BITS EXCEPT LSB
001C C21900             JNZ WAIT        ;IS IT STILL A STØP BIT?
001F CDCA00             CALL DBØUN      ;DEBØUNCE THE INPUT
```

**Figure 2(a)**

```
0022 210C00   L1:     LXI H,12       ;START AT 12 FØR ØVERHEAD
0025 23       L2:     INX H          ;CØUNTER INCREMENTED EACH CYCLE
0026 7F               MØV A,A        ;USED TØ PRØVIDE 5 CLØCKS
0027 DBF5             IN PTIN        ;GET DATA
0029 A2               ANA D          ;MASKS ØFF 7 MSB (REG D = 1)
002A BB               CMP E          ;CHECK FØR A CHANGE
002B CA2500           JZ L2          ;NØ CHANGE YET
002E 5F               MØV E,A        ;SAVE NEW BIT
002F E5               PUSH H         ;SAVE CYCLE CØUNTER VALUE
0030 CDCA00           CALL DBØUN     ;DEBØUNCE THE INPUT
0033 05               DCR B          ;CØUNT THIS BIT
0034 C22200           JNZ L1         ;GØ TØ NEXT BIT
0037 E1       AVG1:   PØP H          ;GET LAST CYCLE VALUE
0038 0603             MVI B,3        ;USE TØ CØUNT NEXT 3 VALUES
003A D1       AVG2:   PØP D          ;GET ANØTHER VALUE
003B 19               DAD D          ;ADD TØ PREVIØUS SUM (16 BITS)
003C 05               DCR B          ;FØR NEXT VALUE
003D C23A00           JNZ AVG2       ;GET REMAINING VALUES
              ;
              ;TWØ SHIFTS ARE USED FØR AVERAGE (DIVIDE BY 4)
              ;AND ØNE SHIFT IS USED TØ DIVIDE THE VALUE BY 2
              ;TØ GET 1/2 ØF THE BIT TIME CØUNTER VALUE.
              ;
0040 0603             MVI B,3        ;USE TØ CØUNT SHIFTS
0042 B7       AVG3:   ØRA A          ;CLEAR CARRY
0043 7C               MØV A,H        ;GET MØST SIGNIFICANT BYTE
0044 1F               RAR            ;DIVIDE BY 2
0045 67               MØV H,A        ;SAVE REMAINING BITS
0046 7D               MØV A,L        ;GET LEAST SIGNIFICANT BYTE
0047 1F               RAR            ;DIVIDE BY 2
0048 6F               MØV L,A        ;SAVE NEW VALUE
0049 05               DCR B          ;CØUNT THIS SHIFT
004A C24200           JNZ AVG3       ;GØ DØ ALL 3 SHIFTS
004D 220013           SHLD DLYCT     ;SAVE 1/2 CYCLE VALUE IN RAM
0050 0E55             MVI C,'U'      ;LØAD 'U' FØR ØUTPUT RØUTINE
0052 CD9F00           CALL DAØUT     ;ØUTPUT (ECHØ) THE 'U'
0055 C9               RET
              ;
              ;THE FØLLØWING RØUTINE RECEIVES SERIAL DATA AND
              ;TRANSFØRMS IT INTØ AN 8 BIT PARALLEL BYTE.
              ;THE INPUT DATA WILL BE IN REG C.
              ;BIT 0 ØF INPUT PØRT IS USED FØR SERIAL DATA INPUT.
              ;
0056 010008   DAIN:   LXI B,800H     ;SET REG B = 8, CLEAR REG C
0059 DBF5     DAIN2:  IN PTIN        ;GET DATA BIT
005B E601             ANI 1          ;TEST LSB ØNLY
005D C25900           JNZ DAIN2      ;WAIT FØR START BIT
0060 CD8200           CALL DLAY1     ;DELAY 1/2 BIT TIME
0063 DBF5             IN PTIN        ;VERIFY START BIT
0065 E601             ANI 1          ;TEST LSB AGAIN
```

**Figure 2(b)**

```
0067 C25600            JNZ DAIN        ;IF NØT START BIT, TRY AGAIN
006A CD8A00   DAIN3:   CALL DLAY2      ;DELAY ØNE BIT TIME
006D DBF5              IN PTIN         ;GET NEXT BIT
006F E601              ANI 1           ;MASK ØTHER 7 BITS
0071 B1                ØRA C           ;GET PREVIØUS BITS
0072 0F                RRC             ;RØTATE B0 TØ B7
0073 4F                MØV C,A         ;SAVE ACCUMULATED BITS
0074 05                DCR B           ;CØUNT 8 BITS
0075 C26A00            JNZ DAIN3       ;GET NEXT BIT
0078 CD8A00            CALL DLAY2      ;DELAY FØR 2 STØP BITS
007B CD8A00            CALL DLAY2
007E CD9F00            CALL DAØUT      ;ECHØ CHARACTER BACK TØ TERMINAL
0081 C9                RET
              ;
              ;THE FØLLØWING ARE THE DELAY RØUTINES
              ;
0082 F5       DLAY1:   PUSH PSW        ;SAVE ACC AND FLAGS
0083 E5                PUSH H          ;SAVE H AND L
0084 2A0013            LHLD DLYCT      ;GET 1/2 CYCLE VALUE
0087 C39200            JMP DLAY
008A F5       DLAY2:   PUSH PSW        ;SAVE ACC AND FLAGS
008B E5                PUSH H          ;SAVE H AND L
008C 2A0013            LHLD DLYCT      ;GET 1/2 CYCLE VALUE
008F 2B                DCX H           ;THESE 2 DCX'S CØMPENSATE
0090 2B                DCX H           ;  FØR SUBRØUTINE ØVERHEAD
0091 29                DAD H           ;DØUBLE THE 1/2 CYCLE VALUE
0092 7C       DLAY:    MØV A,H         ;GET MS BYTE
0093 B5                ØRA L           ;GET LS BYTE
0094 CA9C00            JZ DØNE         ;ALL BITS ZERØ?
0097 00                NØP             ;TØ ADD 4 CLØCK CYCLES
0098 2B                DCX H           ;CØUNT THIS CYCLE
0099 C39200            JMP DLAY        ;LØØP AGAIN
009C E1       DØNE:    PØP H           ;RESTØRE H AND L
009D F1                PØP PSW         ;RESTØRE ACC AND FLAGS
009E C9                RET
              ;
              ;THE NEXT RØUTINE CØNVERTS AN 8 BIT WØRD (BYTE) INTØ
              ;A SERIAL DATA STREAM.
              ;THE BYTE TØ BE ØUTPUT IS IN REG C.
              ;BIT 0 ØF ØUTPUT PØRT IS ØUTPUT DATA BIT.
              ;
009F 0608     DAØUT:   MVI B,8         ;BIT CØUNTER
00A1 AF                XRA A           ;CLEAR ACCUMULATØR
00A2 D3F4              ØUT PTØUT       ;ØUTPUT START BIT
00A4 CD8A00            CALL DLAY2      ;WAIT 1 BIT TIME
              ;30 CLØCKS FØR TIMING BALANCE
00A7 C3AA00            JMP $ + 3
00AA C3AD00            JMP $ + 3
00AD C3B000            JMP $ + 3
00B0 79                MØV A,C         ;GET BYTE
```

Figure 2(c)

```
ISIS 8080 MACRØ ASSEMBLER, V1.0                    PAGE 4


  00B1 D3F4        NEXTB:  ØUT PTØUT          ;ØUTPUT DATA BIT
  00B3 CD8A00              CALL DLAY2         ;DELAY 1 BIT TIME
  00B6 0F                  RRC                ;RØTATE TØ NEXT BIT
                   ;16 CLØCKS FØR TIMING BALANCE
  00B7 00                  NØP
  00B8 00                  NØP
  00B9 00                  NØP
  00BA 00                  NØP
  00BB 05                  DCR B              ;SEE IF ALL BITS ØUTPUT
  00BC C2B100              JNZ NEXTB          ;IF NØT, GET NEXT BIT
  00BF 3E01                MVI A,1            ;SET STØP BIT
  00C1 D3F4                ØUT PTØUT          ;ØUTPUT STØP BIT
  00C3 CD8A00              CALL DLAY2         ;DELAY FØR 2 STØP BITS
  00C6 CD8A00              CALL DLAY2
  00C9 C9                  RET
                   ;
                   ;THE FØLLØWING RØUTINE WAS ADDED TØ PRØVIDE
                   ;APPRØXIMATELY 200 MICRØSEC DELAY FØR INPUT
                   ;DATA FILTERING.
                   ;
  00CA 261A        DBØUN:  MVI H,26
  00CC 25          DB1:    DCR H
  00CD C2CC00              JNZ DB1
  00D0 C9                  RET
  0000                     END
```

Figure 2(d)

Two other extra groups of instructions have been added to accommodate the multiple use of the delay routine. The delay function is called from both the data input and data output routines. Since the data input routine requires more instructions, it was necessary to add extra instructions to the output routine. To illustrate, the DAIN3 loop (6AH to 75H) takes 62 clock cycles, excluding the DLAY2 subroutine. In the DAOUT routine, the start bit timing (A2H to B0H) required only 32 clock cycles until the three "dummy" jump instructions were added for a total of 62 clock cycles. Also, the NEXTB routine (B1H to BCH) used only 46 clock cycles until the four NOP instructions were added.

At L1 (22H), the cycle counter is preset to 12 to compensate for program overhead. Remember, the loop that is being counted consists of 38 clock cycles but when the bit changes state, extra clock cycles are encountered, so the cycle counter must indicate those cycles. The instructions from 2EH to 34H, including the debounce time, consist of 448 cycles. Ten more cycles are added at 22H for a total of 458 clock cycles. Divide this by 38 to get the 12 that is used as the preset value. The first INX H (25H) then increments the counter to compensate for the 38 extra cycles since the last input instruction (27H) at which time the bit change was detected.

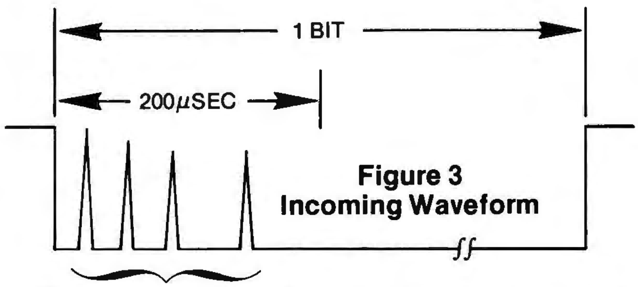



**Figure 3**
**Incoming Waveform**

One other feature that has been added to the program that had not previously been discussed is the verification of the start bit (63H to 67H). This is desirable so that noise pulses that might be wide enough to be detected during an input function (i.e., at 59H), will not cause erroneous reading or writing of data. Although this does not guarantee that no unwanted signals will be detected as start bits, it does greatly reduce the chances of random noise looking like a start bit.

### Input/Output Ports

Since the input/output ports in the SDK-80 are implemented with a programmable device (the Intel 8255), a set-up instruction (3H to 5H) is required to initialize the I/O. The ports are then addressed by means of two address bits, A0 and A1, from the 8080. The other six of the eight bits available for an I/O address are used for chip select of individual devices. This addressing technique is referred to as "linear select" and requires no additional decoding of the address lines. The limitation of this approach is that only six devices may be implemented before a decoder will be required—but this allows as many as 144 I/O lines when multiple 8255's are used. The truth table used in determination of the I/O port address is shown in Figure 4.

| *ADDRESS BIT** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *7* | *6* | *5* | *4* | *3* | *2* | *1* | *0* | *I/O DEFINITION* |
| 1 | 1 | 1 | 1 | 1 | 0 | x | x | USART ON SDK-80 |
| 1 | 1 | 1 | 1 | 0 | 1 | x | x | 1ST 8255 ON SDK-80 |
| 1 | 1 | 1 | 0 | 1 | 1 | x | x | 2ND 8255 ON SDK-80 |
| 1 | 1 | 0 | 1 | 1 | 1 | x | x | I/O DEVICE #4 |
| 1 | 0 | 1 | 1 | 1 | 1 | x | x | I/O DEVICE #5 |
| 0 | 1 | 1 | 1 | 1 | 1 | x | x | I/O DEVICE #6 |

*The address bits A8 through A15 will be identical for I/O instructions.

**Figure 4**
**Linear I/O Port Addressing**

Since the philosophy of this column is to illustrate programming techniques and provide ideas for using a microcomputer, the output port implementation in this example was not optimized. However, in most microcomputer applications it is desirable to make use of as many output lines as possible without having to add extra components. As you will notice from this example, one 8-bit port was dedicated to serial output; only one bit of output port A was used but, due to the approach used, the other 7 bits were not available for any other program function.

In order to resolve this inefficiency it will be necessary to perform some logical operation prior to any output to that port. Further, it requires that the program have access to the current state of all output bits on that port. This can be accomplished by either reading the output port (which the 8255 allows) or by retrieving the previously stored output image from a memory location. Since not all microcomputer systems support the former approach, I will discuss the latter but the logic for either method will be identical.

Wherever an output instruction occurs (A2H, B1H and C1H), additional instructions must be implemented to prevent modification of other bits. These instructions consist of getting the previous data, performing a logical instruction ('AND' for bit reset, 'OR' for bit set), saving the new data, and writing the new word to the output port. Figure 5 illustrates the instructions that would replace an OUT instruction to either clear or set a bit. In cases where it is not known what data is to be output, as at B1H, program steps must be implemented that will test the bit of interest and save the unused bits. Figure 6 shows the steps that will replace the instructions from B1H to B6H. It should be understood that appropriate changes must be made to the routine to compensate for the additional clock cycles used in executing the new functions.

| | *OLD* | *NEW* |
|---|---|---|
| Bit Reset: | XRA A | LDA IMAGE |
| (at A1H) | OUT PTOUT | ANI 11111110B |
| | | STA IMAGE |
| | | OUT PTOUT |
| Bit Set: | MVI A,1 | LDA IMAGE |
| (At BFH) | OUT PTOUT | ORI 00000001B |
| | | STA IMAGE |
| | | OUT PTOUT |

**Figure 5**
**Setting and Resetting a Bit**

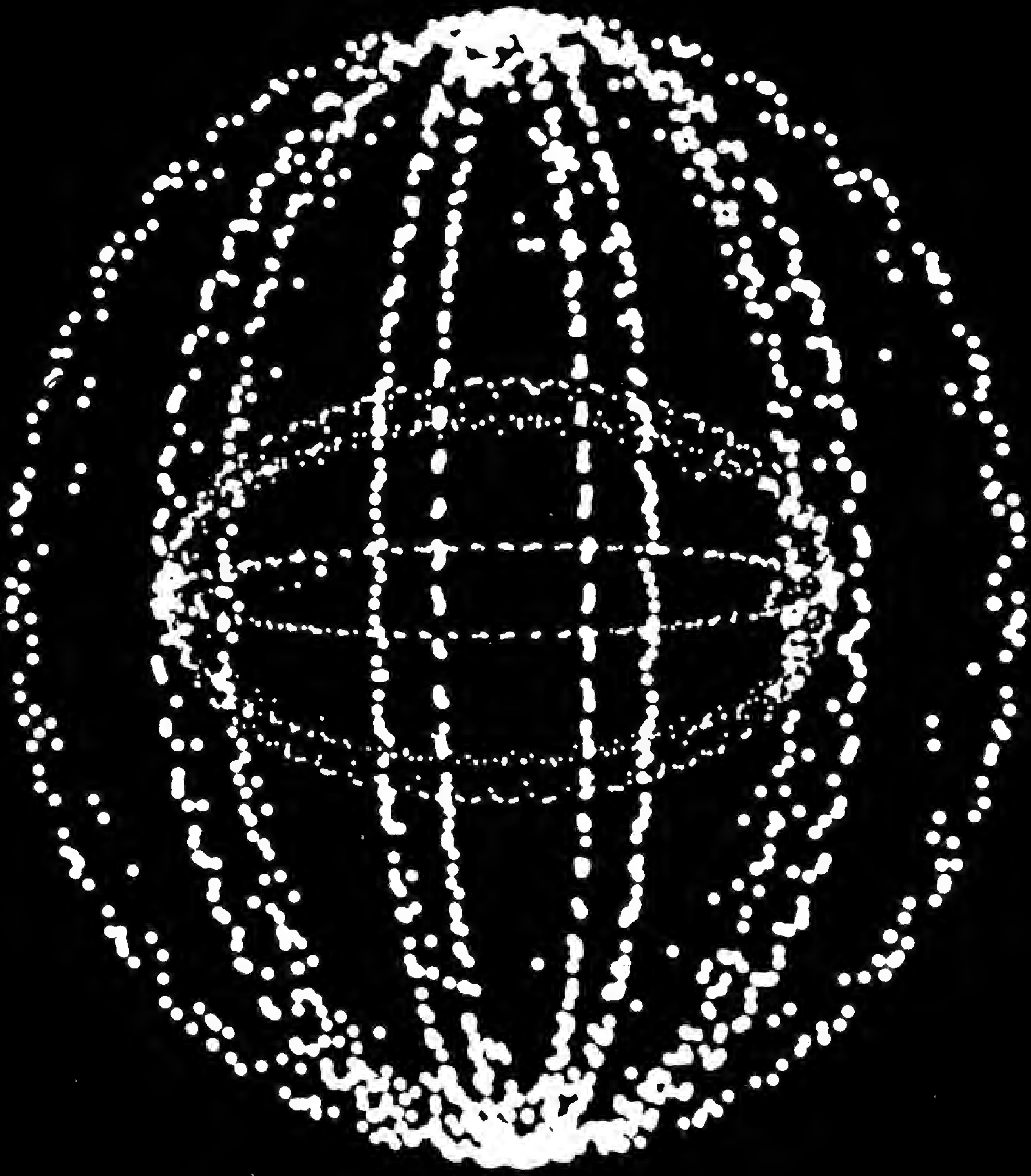
```
;The following instructions can be used to replace the
;OUT, CALL and RRC instructions at B1H, B3H and B6H.

NEXTB:  RRC           ;GET LSB INTO CARRY
        MOV C, A      ;SAVE ALL BITS
        LDA IMAGE     ;GET PREVIOUS OUTPUT PORT
        JC BSET       ;CHECK THE LSB
        ANI OFEH      ;RESET THE OUTPUT BIT
        JMP BOUT      ;GO OUTPUT NEW PORT DATA
BSET:   ORI 01H       ;SET THE OUTPUT BIT
BOUT:   STA IMAGE     ;SAVE FOR NEXT TIME
        OUT PTOUT     ;OUTPUT TO PORT
        CALL DLAY2    ;DELAY 1 BIT TIME
```

**Figure 6**
**Determining Bit Set/Reset Function**

### Summary

The timing loops in this program or any other program are primarily affected by the oscillator and clock generator but are also affected by at least two other portions of a microcomputer system. One is the ready line, as on the 8080, which can be used to implement additional clock periods in cases where slow memories are being used. These clock periods must be included where timing calculations are

Branch to pg. 60



# FIREWORKS
# ANCIENT & MODERN

By John Whitney

Last month I wrote about a visual domain of harmonic pattern and my own theoretical work. Here is more on both subjects. Also I'd like to show how well computer graphics and harmonic pattern are suited for each other and how useful this can be for applying digital graphics to entertain the eye. I referred to the manner by which the composer for centuries has used harmonic "forces" to attract and hold our attention and otherwise entertain the ear. Now let us consider the dynamics of visual "forces."

We can quickly test an elementary dynamic of harmony by halting one note before the last note of any nursery song — we all sense (even the child senses) the incompleteness of the melody. For that matter, just sing do, re, mi, fa, so, la, ti, and observe how strongly 'ti' begs to be fulfilled by 'do' which is the octave above the starting tonic 'do'.

Harmonic theory and practice in music is a complex subject burdened with conflicting attitudes and prejudices which change each generation. But we need only appreciate the above simple demonstration of the power within a nursery tune to realize all we need to know about musical harmony. *In truth, harmonic forces give shape to our experience of past and future.* Just sound a few appropriate tones and automatically expectations are generated within the listener. With occasional spectacular skill, the composer leads us along his own unique musical pathway — rather, his cunning use of harmonic forces push and pull us with such vigor, creating such unexpected expectations, that we may experience ecstatic enthusiasm — or at least we may raise a few goose bumps.

Yet harmonic force is not all that mysterious. We can speculate why the sound of 'ti' urges us on to 'do.' Significantly a good diagram for the perceptual dynamics of harmonics is found in this picture (Figure 1).

It is characteristic of harmonic phenomena, visual, aural or otherwise, to show this kind of pattern. As patterns go, the illustration is perhaps explicit in a way that is even more obvious than the aural leading-tone affect of 'ti' upon 'do.'

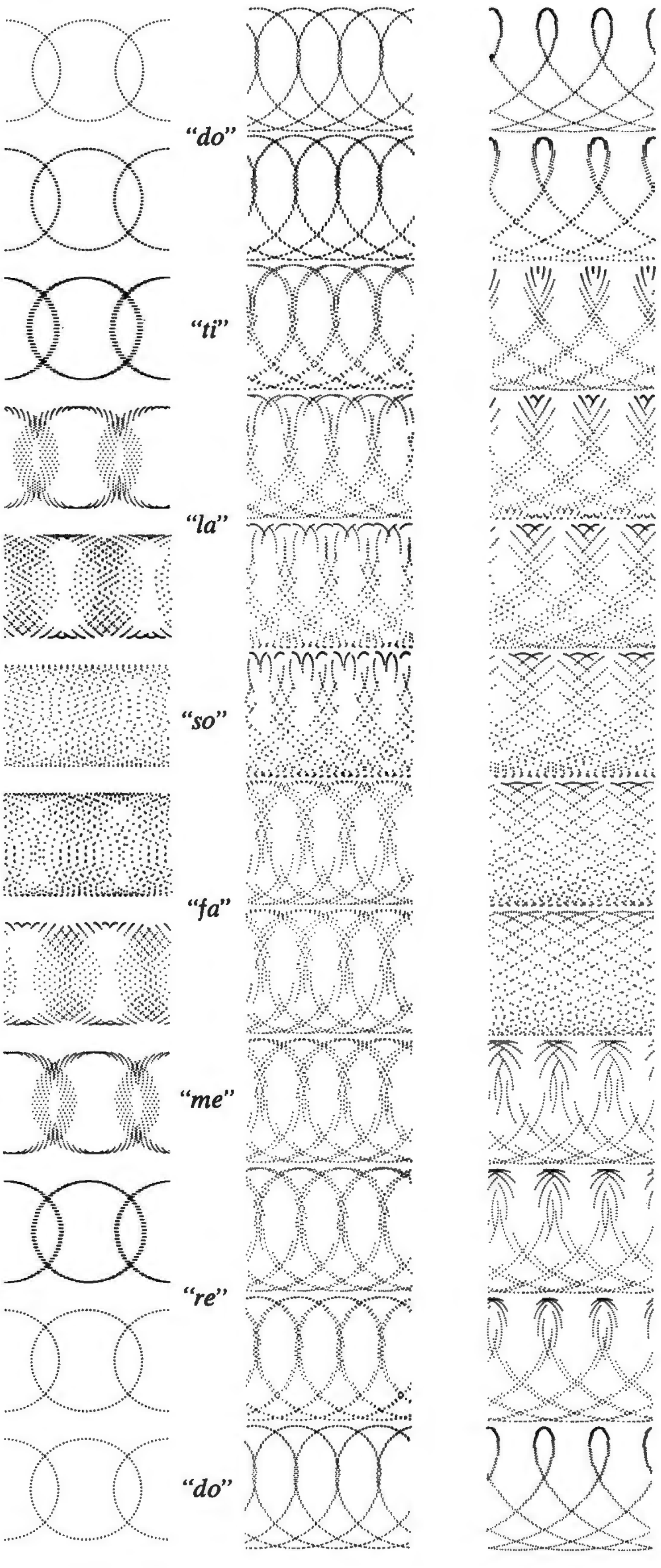



**FIGURE 1.**

*Each of these frames might represent samples approximately one second apart out of a smooth continuous action sequence totaling twelve or more seconds in length. Read from bottom to top.*

Eye and ear, each in its own unique manner, experiences the dynamics of this kind of pattern as an event in time — as punctuation — as an arrival or departure. When we arrive at 'do', the octave above the tonic 'do' we hear that rudimentary relationship with a particular infallibility. If we sample ascending steps of the scale, the ear is bound to sense the final event of arrival just as the eye can see arrival and departure relationships in the illustration. I might add that these relationships are many times more explicit when seen as a motion picture sequence.

The next illustration (Figure 2) is from the first film I made to test the hypothesis that a music-like feeling of resolve could be produced by visual operation upon the mathematics of integer harmonic periodicity. The illustration includes frames at the departure, steps along the way and the final arrival event from that film. The general pattern is derived from a polar coordinate equation (see proceedings Fall Joint Computer Conference 1968, pp. 1299-1305) within a computer program devised by Dr. Jack Citron at the outset of my research grant from IBM. The film is a play upon the idea of arrival and departure with several intermediate punctuations along the way. Most people who view the film agree that it works as conceived. One does anticipate and respond to the final unwinding, decelerating formation of the perfect equilateral triangle symmetrically located in the projection field. Thus the hypothesis had received some substantiation. Substantiation sufficient at least to sustain my search for still deeper understanding.

Another level was achieved a few years later when the film illustrated in Figure 3 was completed. This film is a basic schematized demonstration, if any can be found, of a raw first principle of visual harmony.

Twenty-four hexagonal outline figures are nested in a concentric array. By computer program they are animated to move around a circular pathway at specific rates. In fact, the largest hex figure moves twenty-four times faster than the smallest. That is, in one complete cycle, hex number one (smallest) orbits the circle once, hex two orbits the circle two times, hex three completes three orbits and so on up to hex number twenty-four which completes twenty-four excursions around the circle.

At one-fourth of the complete cycle, the hex figures distribute themselves into four precise and equal groups. At one-third around they form three groups. Halfway, they are two groups. So throughout the complete sequence all fractions group accordingly. There are bound to be four groupings of four each, five groups of five, six groups of six and so on.

In terms of visual perception, vaguely a corollary to aural responses, we have here a phenomena of hierarchical distribution and classification of elements into an array in which all are ranked according to some perceptual scale of complexity. No need to argue whether the unison or the two-part groups are more "consonant" than groups of three or four. For generating dynamic pattern here is an order-disorder principle or a consonant-dissonant principle to work with. It is a principle which can be exploited in more ways than one might expect to give meaningful order to temporal development. It becomes a composer's valid strategy; probably the first strategy to be so defined and applied in the brief history of the art of computer graphics.

Finally, it is worth remarking that none of the illustrations for this article could be created by conventional hand drawing techniques. At least that would be quite difficult. Moreover, it would be impossible to hand-animate the films from which the illustrations were derived. Each of these films required thousands of drawings while the computation for plotting is staggering. Thus the computer is the ultimate and the only tool for visualizing the dynamic world of harmonic functions. This may serve to illustrate the point that this new world of visual art cannot be confused with any previous traditional forms. (See IFIP proceedings 1971, pp 1382-6.)

It should be of particular interest to realize that computer graphics — this 15 year old infant — is patently capable of bringing forth a totally different kind of visual experience as unique and riotously enjoyable — much cheaper — more energy/materials intensive than the Chinese pre-Christian

Branch to pg. 63

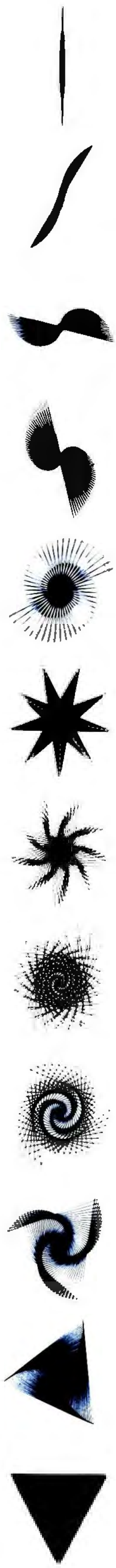


**FIGURE 2.**

*These are sample frames selected from two films, each totaling three minutes in length.*

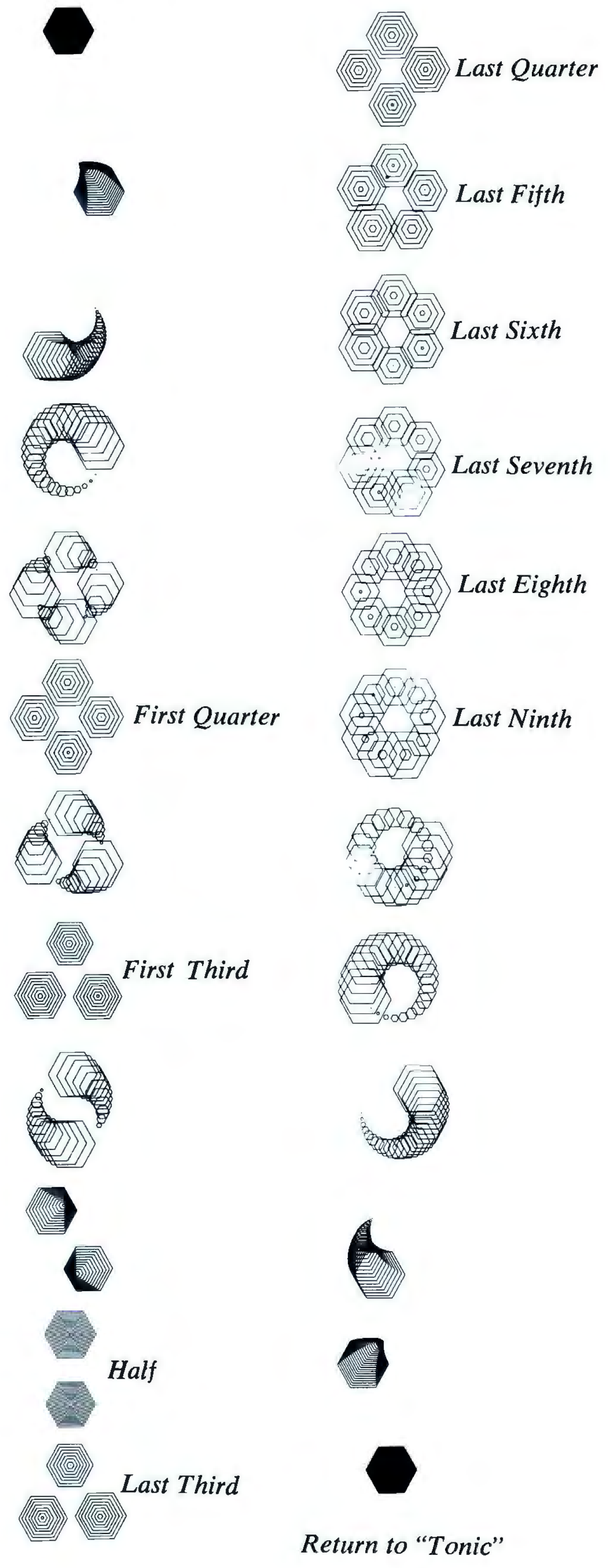



**FIGURE 3.**

*These are sample frames selected from two films, each totaling three minutes in length.*

# The Computer

THE BEST NEWS
LARGEST INVENTORY ON THE WEST

Volume 1 ✱✱✱

# LEAR SIEGLER OFFERS

NEW LEAR SIEGLER KIT

## COMPUTER MART *TO MARKET* ADM-3 KIT

ANAHEIM, CA.—Lear Siegler Electronic Instrumentation Division announced today that it is going to produce its popular ADM-3 Dumb Terminal in kit form. This unique data entry terminal provides full or half duplex operation at 11 selectable data rates, 1920 character display in 24 × 80 organization.

In applications now using teletypewriters, the ADM-3 is software and hardware compatible. The ADM-3 contains both a 20Ma current loop interface and the EIA standard RS-232C interface for direct connection to any standard computer port.

Computer Mart of Orange, Calif., is the first "Full Service" distributor to enter the market (Home Computing) with the ADM-3 in both assembled and kit configurations.

The ADM-3 Terminal Assembled
12 Line $995
24 Line $1175

The ADM-3 Terminal Kit
24 Line $875
(12 Line not available in kit)
Lower Case Display option $99

*ADM-3 DUMB TERMINAL*

## *THE COMPUTER DOCTOR*

# COMPUTER MART SERVICES

ORANGE, CA.—The Computer Mart, serving the West Coast for off-the-shelf personal computing needs, offers unique services to manufacturers and customers alike.

"We are committed to full service and support to our customers and manufacturers," says John French, General Manager of Computer Mart. "If one of our customers has difficulty with his system, software or kit, he can bank on our help to get it up and running."

George "The Computer Doctor" Tate, Service Manager at the Computer Mart, supplies customers with technical back-up when needed and feedback to the manufacturer to ensure rapid solutions to design or operational pitfalls.

The Computer Doctor services *all* personal computing systems—not only those sold by the Computer Mart.

CIRCLE NO. 11 ON INQUIRY CARD

# Mart Gazette

ALL THE TIME
COAST FOR PERSONAL COMPUTING PRODUCTS

May 1976 (714) 633-1222

# ADM-3 KIT-LOW $$

## ADM-3 KIT EASY TO ASSEMBLE

*Busy at Work Assembling*

ORANGE, CA.—New kit builders or old timers will find the ADM-3 extremely easy to assemble. The kit comes as two basic assemblies: CRT section pre-mounted in cabinet, and keyboard control and display electronics section with power supply pre-mounted. In-depth technical manual supplied is excellent. Layout design of the kit utilizes a modular building block concept allowing maximum construction simplicity for the builder. All integrated circuits are socketed and therefore easily replaceable. It is important to note this major advantage exists only in the kit configuration. All components supplied are fully guaranteed with no warranty implied, expressed or extended to any terminal assembled from this kit. Technical support is available from The "Computer Doctor" at the Computer Mart, should any difficulty arise.

COMPUTER MART FIRST AGAIN

## *ANOTHER* COMPUTER MART *FIRST*

ORANGE, CA.—Computer Mart has been instrumental in developing new markets for companies entering the personal computing arena. The Computer Mart was the *first* IMSAI distributor providing this advanced 8080 microcomputer system to its clientele. Processor Technology became the second company to choose The Computer Mart as its *first* distributor providing memory & peripheral boards.

Today, the list of companies has grown to include: Solid State Music Co.—static memories, miscellaneous boards; Cromemco—TV Dazzler, 8K Bytesaver® programming aid; Sanyo Electric—9″, 13″ and 15″ B/W TV monitors and 19″ color monitors; Texas Instruments—I.C. sockets and connectors; Vector Electronics—PC boards, wire wrap tools and supplies; Tarbell—cassette interface; Terminal Systems—ASR—33's, rebuilt and guaranteed; and Polymorphic Systems—video display boards and processors.

Computer Mart looks forward to bringing its customers the new Lear Siegler ADM-3 Dumb Terminal kit—the response to many requests.

*W. A. Terry, Lear Siegler and John French, Computer Mart, Meeting the Needs of the Home Computer Market.*

## COMPUTER MART, LEAR SIEGLER OPENING NEW FRONTIERS

ANAHEIM, CA.—Lear Siegler Electronic Instrumentation Division in entering the personal computing field is testing a new direction and philosophy in the company's marketing plans.

Realizing the real and exploding need for low-cost data entry systems for the home, and the need for individualized customer service, they have equipped the distributor with all necessary design and service information so that close customer support may be handled expertly on a local level rapidly.

*Service, Interfacing, Advice and Guaranteed Kit Assembly*

**THE COMPUTER MART**
Authorized Dealer

**625 W. Katella Avenue, No. 10**
**Orange, CA 92667**
**(714) 633-1222**

# TOWARD THE UNDERSTANDING OF LARGE SCALE SYSTEMS

*by Ralph Klestadt*

The structures of operating systems, specifically timesharing operating systems, are often specialized and are often radically different from each other. Because of this aspect of large scale systems, an examination of two major types of timesharing systems is due.

We will be examining three major types of operating systems, all on timesharing computers. The first will be the IBM TSO system as used on the IBM System 360/370 Computers. The second and third systems are used on the DECsystem-10 and DECsystem-20 Computers.

Shown in Figure 1 is a diagram of the IBM TSO (Time Sharing Option) operating system. At the top is the user who maintains control over all under him. Directly below the user is the interconnection system, which is the hardware/software part of the user's connection to the machine. Under TSO, the user must set his parameters and maintain control over the interconnection unit. This is a technique commonly employed on TSO, however this is also commonly modified so the machine handles this step. Directly below the interconnection system is the monitor. The monitor is the main program which contains complete control of its sub-programs. From the TSO monitor, the user has access to two major systems: the utilities and the editors. In order to access the utilities, the user must tell the monitor to place control of his job in a separate program which still runs under control of the monitor, or a utility. Utilities may do different jobs such as list processing or file formatting. While under control of a utility, the user does not have access to monitor commands, such as working on files and determining the status of the system. The user must stay within the bounds that are imposed upon him by the particular utility he is running. This remains constant throughout most timesharing systems. It is the editor that makes the major difference that separates IBM TSO from other systems. While the user is on the monitor, he does not have direct access to editor commands. The user must run the editor program, and then either work on an old file (changing or re-writing it) or create a new file. While on the editor program, the user does not have access to monitor commands such as accounting and opening of new files. The user must leave the editor program in order to do this on TSO. On TSO, before a program may be created, space on the disk must be allocated, which requires the typing of a statement to the monitor which can be somewhat complicated. Inter-user communication on TSO may be somewhat complicated, particularly when it has to be done through a utility called MAIL. This requires going off the monitor level, which is usually annoying when one is creating a file and at the same time typing a message to another user.

*Figure 1*
*IBM TSO*

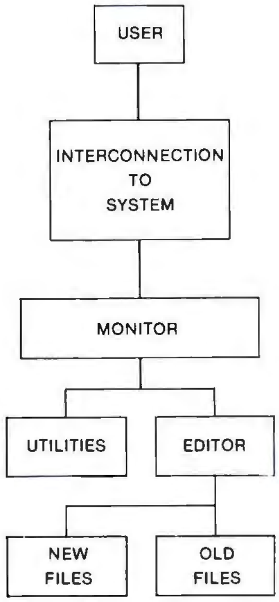
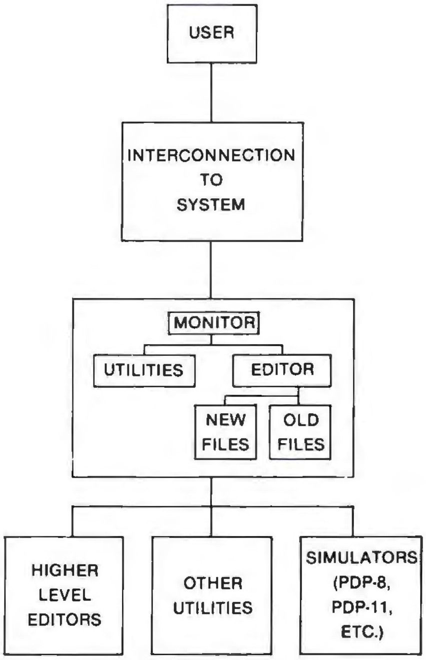



Figure 2 shows a diagram of one of the monitors used on DECsystems. This particular timesharing system is the ICS system, or Integrated Command System. By simply looking at the diagram, you can see marked differences between it and TSO. When the user is on ICS, all interconnection duties are handled by the computer, which in turn makes a logon procedure a two line procedure. In ICS, a lot of the programs that a user must call on TSO, are included as the set of monitor commands. Many utilities are in the command set, such as a command called MONTH. On TSO, the user must run a program to get the accounting information. Yet on ICS, the user simply types MONTH to the computer and gets back all accounting information without ever having to run an external program. ICS also has a built-in editor in the monitor,

which eliminates a lot of the problem experienced on TSO when working on the monitor and editor at the same time. On ICS, the user may open up a new file or edit an old one while staying on the monitor. This way the user may be editing a program and also may be communicating to another user at the same time. The user, however, does have access to other editors, which are much more powerful and must be run externally if he needs advanced editing capabilities. Another nice advantage of the ICS system is that the user has easy access to simulators. Simulators are system programs which emulate other computers, such as a PDP-8 Simulator. It is run under the monitor, but while running the simulator, the big computer acts exactly like the simulated computer. For example, when running the PDP-8 Simulator, you would only be able to use the PDP-8 instruction set. The big advantage of a simulator is that you can debug and run the program on the large system simulator, and then run the finished product on the minicomputer.

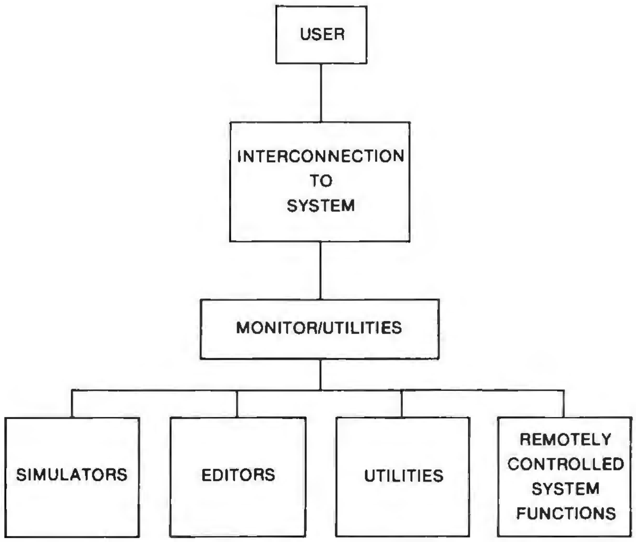
*Figure 2*
*ICS*




When you compare the system of disk storage allocation, there is a marked difference between the two timesharing systems under discussin. In TSO, the user usually has to specify the parameters in the allocation statement. Yet in ICS, the machine will take default specifications for just about everything, unless you specify it to be something different. When allocating many files, this generally saves much time. In general, ICS is easier to use, however it does not offer the wide selection of languages available on IBM's TSO.

Figure 3 is a chart of TOPS-10, the most commonly used monitor on the DECsystem-10. A new version, TOPS-20, which has just been released and is used on the DECsystem-20, is much like TOPS-10 but is more advanced in the area of communications. As in ICS, interconnection to the system is handled by the computer, and logon is the same as in ICS. The major difference between this monitor and ICS is that while ICS offers monitor level editing capabilities and TOPS-10 does not, TOPS-10 offers very sophisticated monitor level system functions, such as a list of all the users on at one time, ability to run remote peripheral devices through the monitor, and a unique operator service program. In TOPS-10, the user has access to advanced utilities on the monitor, much as in ICS. Utilities may also be run externally, and there may be some very advanced utilities not found on other systems such as programs for debugging object level programs. In the field of editors, TOPS-10 offers two main editors. While ICS offers TECO (Text Editor and COrrector), a very advanced editor, and FILGE (FILe Generator and Editor), TOPS-10 offers TECO and SOS (Son Of Stopgap). SOS is a combination of FILGE and ICS, yet different. IBM's editor, called ED for short, is somewhat like TECO in capability and complexity. With any of these editors, proficiency is gained through practice. Under TOPS-10 the user has access to simulators as in ICS, which are usually the same ones or modifications of those used in ICS.

*Figure 3*
*DEC TOPS-10*




One sure way to differentiate between monitors is to see the prompt, or character(s) printed when the computer is ready to accept input. In IBM TSO, the word "READY" is printed. In ICS, 'OK" is printed, and in TOPS-10 a dot (".") is printed.

Actually, the only way to compare monitors is to work on them. Once the user has compared the features and advantages of each, then he can decide for himself which one is best suited for him.

# THE IMSAI 8080

*by Daryll Shatz*

*EDITOR'S NOTE — The following article is an edited composite of thoughts and findings of several people who built some of the early IMSAI units. The major portion of the work is that of Mr. Shatz.*

*We must, however, make the comment that at this time there does exist a small incompatibility with the ALTAIR system. This is in the memory protect circuitry. The IMSAI has no front panel protect/unprotect switch. There is no way to currently protect the contents of a non-IMSAI memory when in an IMSAI. The IMSAI memory boards have a "write protect" slide switch near the bottom of the board which must be moved to the protect position by manipulating a small object between the boards once the board's program has been loaded. We cannot really commend this feature; however, future IMSAI memories are scheduled to have programmatic control of the protect status which, we feel, will have high merit.*

In recent years, there has been a surge of new and exciting products on the electronic market. A few have been original, but most have been a better idea of an existing product. There have always been those who can make something better, and luckily we can all reap the benefits.

The IMSAI 8080 microcomputer system was born of the better idea formula. And it appears that they have succeeded. "They" consists of a small group of individuals who started a hardware/software consulting firm over five years ago. Since then, IMSAI (information Management Services Associates Inc.) has directed their efforts towards the microcomputer market.

The IMSAI 8080 Micro System is a new design of the MITS ALTAIR 8800 microcomputer. The two units use the same central processor chip, the INTEL 8080, and use the same buss structure. Therefore, both units are hardware and software compatible. The most noticeable differences about the IMSAI are the front panel design and the magnificently clean internal wiring. The front panel has an array of color coded plastic switches, rows of LED's, and a unique sandwich of a photo mask legend and two sheets of acrylic plastic. The legend has both octal and hexadecimal formats indicated on it. The internal wiring is held to less than 30 wires, all of which are power supply connections to and from the unit's 24 amp supply (another 24 amp supply drops in when needed).

Now that you know what the IMSAI 8080 is, let's look at how it is put together.

Unpacking the kit, one finds four large packs containing the printed circuit boards and their parts, two reference manuals: AN INTRODUCTION TO MICROPROCESSORS (400 pages), INTEL'S 8080 MICROPROCESSOR SYSTEMS USER'S MANUAL (200 pages), and IMSAI's three ring binder Systems Manual.

Normally you begin construction of a kit by reading the manual. However, this kit is an exception—you have to assemble the manual first. The manual is quite complete and is divided into logical sections for chassis, motherboard, power supply, MPU, etc. Each section contains parts lists, assembly instructions, users guide, theory of operation, schematics and drawings.

After reading the manual, construction begins with the chassis and power supply. The power supply is designed around a hefty transformer, three giant capacitors, and a small p.c. board. The chassis goes together with self-tapping screws.

The front panel is really a plug-in circuit board consisting of the 24 color coded static proof switches, legend sandwich, and components. The most difficult portion of this construction is getting the 44 LEDs aligned, which is simplified with a small homemade jig.

Probably the most boring part of the entire construction is that of the motherboard. With the optional 22 slot motherboard, there are 2200 tedious solder connections to be made for the socket pins. The standard unit comes with only a six slot motherboard and three sockets. After the mother board is finished, it must be screwed down to a metal mounting bracket. However, once this is done, the pins of the 100 pin connectors and the mounting bracket are extremely close to shorting out. This can be fixed with extra washers between the motherboard and the mounting bracket.

The processor board is designed around the INTEL 8080A CPU, the 8224 clock generator, and an 8212 data latch for storage of the CPU status signals. All signals out of the CPU and the buss drivers are TTL compatible.

The 4K RAM board is designed around INTEL's 8111/2111 memory chips. The board comes with only 1K but is expandable to the full 4K with only the addition of more memory I.C.'s.

The final phase of the assembly is the checkout. First all voltages are checked at both the power supply and on each card (with the 8080A removed). Once the voltages check out, the 8080A chip is inserted and the test programs given with the unit are run as a final verification on its complete operation.

Components in the kit are of a very high quality and the p.c. boards are neat and well laid out. Construction, for the experienced kit builder is straightforward. However, Heathkit fans will be disappointed as the instructions are not given component by component. Rather, the instructions will say something like ". . . the 24 integrated circuits can be inserted at this time . . ."

Once built, the IMSAI 8080 is a high quality microcomputer system, a good example of the "Better Idea".

# The President's Column

The Southern California Computer Society will hold its annual election for Officers and Directors in September. As provided for in the By-Laws, Marv Perlman has been appointed Chairman of the Nominations Committee. I strongly encourage anyone who is willing to serve as an Officer or Director to submit his or her name to Mr. Perlman prior to May 22, 1976. Every potential candidate is encouraged to attend our Board Meetings and decide for himself whether we should all be replaced by a random number generator.

Seriously though, I congratulate the present Board members for an outstanding job in managing the Society's affairs this year. They have devoted vast amounts of time and effort at great personal sacrifice to lay the foundations of a strong, dynamic Society. If any of them is willing to seek re-election, and I hope they all will, I endorse them wholeheartedly. They have represented you, the members, well and ably.

To replace those who want to retire to friends, families and forty-hour weeks, members with courage and determination will be kidnapped, nominated and elected. Are you willing to try?

Our Treasurer and Group Purchase Chairman, Hal Lashlee is returning to private life. His outstanding work in developing and expanding the group purchase activity has saved thousands of dollars for our membership. The fantastic amounts of time Hal has given us is deeply appreciated and the Board of Directors has adopted unanimously an expression of thanks and good wishes. The group purchase activity is now being handled by Chris Leach. Ruben Loshak has been appointed Acting Treasurer until a special election is held.

The Society By-Laws were presented in the April issue of INTERFACE. If you could sense the uncertainty of the small group who adopted them, you might recognize that some people suspected how large the Society was to become. There are some 240 odd Charter Members who agreed to a 20 member Board of Directors. The United States Senate began with only 26 members. Several large corporations have fewer than 20 directors. My point is that the Founders did a magnificent job of adopting laws that allowed the Society to flourish as it has. Very shortly you will be asked to vote on amendments necessary to insure the future of the Society in September. The next administration may well be faced with representing 24,000 members . . . a far more difficult task then this administration began with.

The Board of Directors has authorized two awards for outstanding pioneering efforts in computer hobbyist journalism. The first award is to "Micro-8 Computer User Group Newsletter" and its editor, Hal Singer. The second award is to BYTE Magazine and its editor, Carl Helmers. These two publications represent the upper and lower limits of the spectrum of national information sources (with INTERFACE hopefully somewhere between). My personal favorite is "Micro-8" because of the direct person-to-person contact that comes through complete with spelling errors. A second reason is that, being nearsighted, I can read it more comfortably with my glasses off.

Now, as to the award to BYTE, I find that difficult to understand. Ever since I found a copy of the magazine in my dentist's waiting room I have been suspicious of them. Any magazine whose first issue is about to be listed on the New York Stock Exchange (17 ¼ bid - 17⅝ asked) has to be carefully watched. Nevertheless, the Society has seen fit to award BYTE the sterling silver combat helmet and casserole combination in recognition of not doing too bad a job at whatever it is they do.

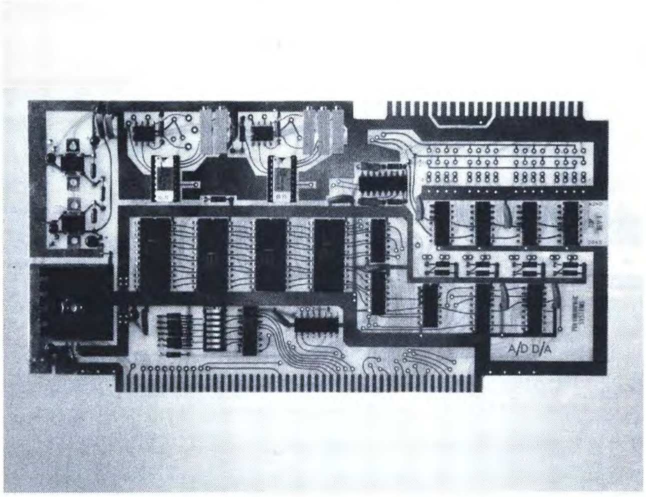
I apologize for taking up so much of your time with Society business. If it were possible not to, I would much rather devote this space to what to do when you discover that you have assembled your IMSAI power supply board upside down.

Respectfully submitted,
Ward Spaniol
President

# Polymorphic Systems Catapults Into Technological Gap

*by Sheila Clarke*

What do you do when, after purchasing and assembling your Altair, you discover a technological gap in the hardware? How do you solve your need, for instance, for an A/D-D/A interface? If you're lucky, you meet up with someone who has the capability to bridge the gap. That's just about how it happened when Rich Petersen and Brian Wilcox decided to go in together to jointly assemble the Altair Computer, then an innovation to the computer hobbyist. Rich and Brian were both working as computer systems analyists at a company involved in government transportation research. As the course of events will take some, they met a co-employee, John Stephensen who, some months previously, had completed his development of an 8080 base system from scratch. The three decided to join forces to further augment their pet computer.




*(Photo: A/D D/A interface board)*
*The Altair compatible A/D D/A interface board currently Polymorphic's bread and butter, allows the home computerist to interface his computer to the outside world. Many devices hook up to the Altair or MICRO-ALTAIR™ and have analog inputs or outputs. This board is a complete general purpose interface that provides 1 or 2 analog outputs with 10 bits of resolution and 6 digital control lines. The 8 analog inputs may be used as programmable level sensors, null detectors or software driven analog to digital converters, and any or all may be selected by on-board jumpers.*
*A typical application for the device is to use the 2 analog outputs to drive the X and Y channels of an X-Y plotter. Digital outputs are used to control the pen and turn the plotter on or off while 4 of the analog inputs are used as feedback null detectors. The null detector outputs may also be connected to the Altair interrupt system. The company provides application notes with each board describing the interface to most analog I/O devices and sensors, as well as listings of several I/O routines.*


*Having been hobbyists themselves, the three men decided to produce a piece of equipment designed to more perfectly meet the needs of the home computerist by making expansion a reasonable alternative. This video interface plugs into the MICRO-ALTAIR™ or IMSAI 8800 bus and connects to a standard TV monitor, or modified TV receiver. The 8-bit input connects directly to most keyboards. Character set includes 128 upper and lower case ASCII and 64 graphics characters for plotting 48 × 128 (or 64) grid. Characters are stored in the onboard memory and may be accessed directly by the computer. Output is standard 525 line EIA RS170 video.*

The project went very well, and they had their machine up and running using the Altair, John's 8080 base system, a Decwriter, their newly created A/D-D/A board and finally, their own very versatile video board. Realizing that brother hobbyists also needed help, they began to market the yield of their labors, and the new Analog Board launched Polymorphic Systems into business.

The founders of Polymorphic Systems were still employed, and working at their hobby/profession, when they noticed that the aforementioned gap existed for the home computerist. So the video interface development ensued, was refined and went into production. Business was brisk . . . it was time to take the hobbyist's needs seriously, and as one idea led to another to bridge more gaps, Rich, Brian and John left their secure employment to launch into the business world full time on December 31, 1975.

Six weeks later INTERFACE Magazine visited the new offices of Polymorphic Systems. The three enthusiastic, energetic and bright young men who know they have a lot to offer the home computerist, showed us

through their newly constructed headquarters. On entering we found a well equipped front office, an engineering section beyond, and behind that plenty of room in which to expand (which they'll certainly need to do). The drafting table was neatly laid with the design of their newest circuit board. Next to that sat the Altair, hooked to a DECwriter and a Sanyo monitor. The screen displayed a multi-patterned configuration that demonstrated some of their graphics capability. We walked around to the back and noticed a small computer, posing under lights before a Poloroid camera. The little machine turned out to be Polymorphic's latest contribution . . . the MICRO-ALTAIR, now renamed POLY-88.


*The MICRO-ALTAIR™ is a complete computer, the latest product offered by Polymorphic Systems, which includes video interface board, CPU/ROM/RAM board, backplane with power supply, all compactly packaged in wood grain finish cabinet. All that is required for a ready to use computer system is a keyboard and TV monitor. It's finest feature, other than being complete and small, is expandability. Several of these units may be plugged together serially for system expansion. As each backplane contains its own power supply, system expanison is easily accomplished without worry of overloading the original power supply. The system is compatible with existing peripherals and software supplied by various manufacturers.*

John Stephenson, an electronics engineer who has studied at UCSB, is Polymorphic's hardware man. John also contributes software and assembly language programming. Brian Wilcox, who's background includes Bachelor degrees in physics and mathematics, produces the literature and artwork. SCCS member Rich Petersen, also with a Bachelor degree in physics, designs and packages the products, and is the company administrator. The three envision the day when you, the home computerist who may know little about the hardware needed for your own system, can purchase a neat package that contains more than the basic computer. You'll have I/O interface, sufficient memory, storage, ample power supply, video interface, expansion capability in a choice of directions, and other built-in features that are now offered as options from a number of manufacturers. Polymorphic Systems foresees its systems being utilized in educational environments that will stimulate student learning and make the teachers' jobs not only easier, but far more exciting. Right now the company is most interested in providing sorely needed hardware in an integrated system. It is notable that although they have addressed themselves mainly to the home computerist, at least 50% of their response has been from professionals, large and small businesses, and physicians and accountants who have expressed interest.

Of the three kits offered, the video interface board is expected to be the most popular for the next few months. The company's success seems immanent. At the rate the partners are perceiving problems and envisioning solutions, we hope they can fulfill current needs before catapulting into orbit with systems that surpass anything now available.



*For you sharpies who just* knew *something was missing from the March issue of "Serial Data Com"*

$$C = 10 + 32 + N \times 34 + 19$$
$$= 61 + N \times 34$$

if $N = 0$, then $C = 61$

therefore $T_{min} = 61 \times .5$ usec

$= 30.5$ usec

if $N = 65535$ then $C = 61 + 65535 \times 34$

$= 2228251$

therefore $T_{max} = 1.114$ sec.

To determine N given a desired period:

$$N = \frac{C-61}{34}$$

$$= \frac{T/.5 - 61}{34}$$

where T is in microseconds.

To obtain a period of 4.55 msec, $T = 4550$, then

$$N = \frac{4550/.5 - 61}{34}$$

$$= 265.85$$

# No Such Thing as Cheap Timesharing

*By Alden Rhodes*

You know the story . . . they hook you up at 10¢ an hour and the storage cost is enough to make you go out and refinance your house. Or the cost of an hour has you going out making hourly withdrawals at your friendly neighborhood bank. And the salesman conveniently left out the fact that a $100 deposit is needed before you get the privilege of using the computer that additionally will cost you a $100 per month minimum.

Well! The hour of SCCS TIMESHARING has arrived. One of our members has put together a timesharing service specifically geared to SCCS members and their pocketbooks. SCCS TIMESHARING offers members only BASIC timesharing at 75¢ per hour at a 10 CPS terminal speed. Additional charges break down this way . . .

Computer resource units, 1¢ each
Typical user will run in the range of 25¢ per hour doing interactive program development, debugging and testing. Or doing text editing.

Computer disk storage, 50¢ per month per 1000 characters.

Monthly service charge of $2.70.
No minimum.

Typical computer club member timesharing usage has been . . .

1. Speedy interactive writing, debugging/testing of programs.
2. Backup for the Extended-Super Basic user.
3. Text editing (also known under the term "word processing").
4. Software development, especially related to membership interests such as cross assemblers, simulators, etc. related to chips and products used by the membership.
5. User terminal discounts (either buy or lease).

SCCS TIMESHARING is based on a Nova type computer specifically configured to support an extensive number of users with file handling characteristics, particularly suited for development needs.

The current configuration is set up to handle 128 concurrent users.

Equipment can be added to the system to handle more, should usage warrant. But note: a typical 128 user capability would support 500 to 1,000 short period intermittent users, so an early capacity problem is not likely.

The Basic language handles strings, arrays of strings, direct address and index address files and a randomized type of file that is direct address, but does not require the reservation of space for records that may be created. File will grow and be sized only according to the actual number of records in the file, regardless of the records "record number".

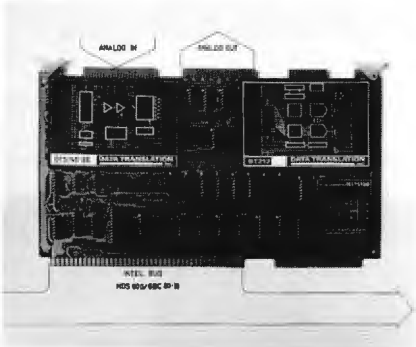
Numerical accuracy is selectable by the user in the range of 4 to 14 digits which is a range of $\pm 7999$ to $\pm 99999999999999 \times 10^{\pm 63}$.

Not to belabor the point . . . the SCCS member can now set his hands on a computer with a basic language similar to what he may actually wind up purchasing and/or assembling. He can get the feel for computer action before his equipment is finished. And he can utilize timesharing as an extension of the capability of his own equipment. SCCS TIMESHARING users are not expected to be BIG spenders, nor are they encouraged to do so. Rather the members' budget will be treated gently and with respect. SCCS TIMESHARING offers the small and medium BASIC user facilities priced to his advantage.

If you are potentially interested in taking advantage of this service, please contact us at SCCS TIMESHARING (213) 629-5468.

P.S. There has been a lot of talk around SCCS meetings about wanting some low cost timesharing. This is not a commercial system, but only offered on these terms to SCCS members. If there is not significant membership interest SCCS TIMESHARING will be dropped. So let's make the membership interest (or lack thereof) known.

# New Products




CIRCLE NO. 90 ON INQUIRY CARD

## SINGLE BOARD ANALOG I/O SYSTEM MATES WITH NEW INTEL SINGLE BOARD MICROCOMPUTER

Framingham, Mass—A complete single-board analog input and output system, built to be exactly compatible with Intel's new single board SBC 80/10 microcomputer, has been announced here by Data Translation, Inc.

The new system, the DT1751, offers a 16 channel high speed data acquisition system, at the input, 2 Digital to Analog converter channels at the output, and a program I/O and interrupt interface to the SBC 80/10, as well as to Intel's MDS-800. The Intel microcomputer together with the DT1751 form a complete computerized data acquisition and analog output system for monitoring and control applications in industrial and laboratory processes. The DT1751 is available in two weeks ARO and is priced at $795 in 100 quantity.

According to Paul Severino, the company's director of computer products, the combination of the single board analog I/O and the single board computer "offers a standard compact approach of bridging the gap between microcomputers and analog measurements. For under $1100 a complete analog input/output computer system can be implemented. Dedicated remote analog measurements can be made at low cost and the data sent serially to the host computer. This method completely off loads the main computer from the burdensome I/O tasks."

The DT1751 analog I/O board is mechanically and electrically compatible with the Intellec™ MDS-800. Application software can be developed in the MDS-800 system and subsequently run with the SBC 80/10 in OEM applications. Further the interrupt portion of the interface has external trigger capability for synchronizing data measurements with some process event.

All analog circuitry within the DT1751 utilizes standard DATAX modules. At the input is a DATAX II data acquisition module, complete with an over-voltage protected MUX, Sample/Hold, and 12 bit A/D Converter. An added feature is a programmable gain amplifier for extending resolution of analog measurements to 14-bit resolution. The analog outputs comprise two channels of high current drive 12-bit D/A converters, capable of driving 50 feet of cable to 0.1% in less than 1 usec. Applications where actuators and servo amplifiers must be controlled, can be driven directly with these D/A converters. Additionally, a Z-axis control is included for the analog outputs for point plotting applications of CRTs, pen plotters, and XY recorders.

The DT1751 is packaged on the same small-sized card as the SBC 80/10 6.75" × 12". Height of the card is 0.375" for accommodation in standard 0.5" spacing common to microcomputer systems. The unit price for the DT1751 is $1195.

Data Translation is a leading developer of new microprocessor and minicomputer compatible analog I/O equipment for measurement and control applications.

For further information contact Fred Molinari, President, Data Translation Inc., 109 Concord St., Framingham, MA 01701; (617) 879-3595.


CIRCLE NO. 91 ON INQUIRY CARD

## MFE CORPORATION INTRODUCES BUFFERED DATA TERMINAL SYSTEM

MFE CORPORATION, a Salem, New Hampshire, computer peripheral manufacturer, today announced its first product entry into the communications terminal market.

The MFE Model 5000 Buffered Tape Cassette Terminal is built around MFE's most successful and reliable, field-tested Model 250 Tape Cassette Transport System. To date, more than 15,000 Model 250's are in use. Controlling the buffering, editing, and communications is the integration by MFE of the Intel 8080A Microprocessor.

"The Model 5000 is the only system to be introduced to this market which utilizes a micro-computer for control and efficiency," said MFE's Model 5000 Product Manager, Maury Kirby. "Programmed high speed search and powerful edit functions as well as full communications capability are all standard features of this highly reliable terminal system."

"In the past, the user had to select from long, confusing option lists in order to tailor a terminal to meet his varied application requirements. Now configurating is made easy and with the highly reliable Model 250 and 8080, the OEM and end-user can benefit from error-free data preparation and transmission."

"The Model 5000 has the best price/performance of any previously installed or announced buffered tape cassette terminal in the industry. Quantity prices for the OEM, end-user, and distributor are available."

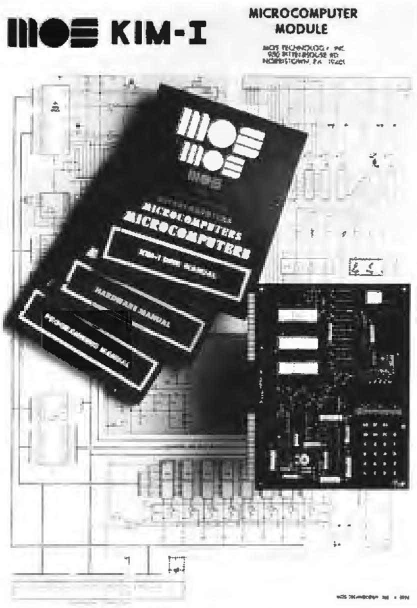
The MFE Model 5000 conforms to ANSI/ECMA standards, (Industry Defined), with tape speeds of up to 120 inches per second on cassettes capable of holding 155,000 characters of formatted data.

The system is compact, weighing less than ten (10) pounds and is adaptable to universal power and environmental conditions.

MFE CORPORATION will begin production of the Model 5000 in March from its 70,000 square foot manufacturing facility. MFE is now in its fifteenth (15) year of operation, and has expanded to 250 employees. MFE CORPORATION markets its diversified products throughout the world.

Contact Mr. Maury Kirby for pricing information.

M.F.E., Keeway Drive, Salem, New Hampshire 03079; (603) 893-1921.




CIRCLE NO. 92 ON INQUIRY CARD

## KIM-1 A COMPLETE MICROCOMPUTER SYSTEM FOR $245

MOS Technology, Inc. has begun deliveries of the KIM-1 Microcomputer System. The unit is supplied at a cost of $245 and includes a single module, full documentation, and all monitor and operating programs stored in ROM.

KIM-1 is NOT a kit! The system is completely assembled, fully tested, and warranted against all defects.

Included in the KIM-1 System are:

—A 605 Microprocessor Array.

MOS Technology's recently introduced N-channel, 8 bit microprocessor with 16 bit address range, 13 addressing modes, a powerful instruction set, and multiple interrupt capability.

—Two 6530 Arrays.

Each 6530 array provides 1024 ROM bytes, 64 RAM bytes, 15 I/O pins, and an interval timer. The monitor programs are stored in the 2048 total bytes of ROM.

—A 1024 Byte Static RAM

—A 24 station keyboard

—A 6 digit LED display

—All interface circuits and logic for operation with TTY or other serial teleprinters. Baud rates from (110 to 1200 baud) can be used. The system adjusts

automatically for any data rate.

—A foolproof interface to low cost audio cassette tape units for recording and playback of programs and data.

—A 1MHz crystal controlled system clock

—15 bidirectional I/O pins for control of a specific application

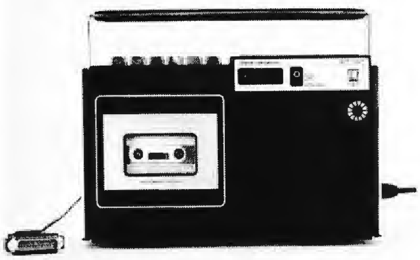
The unit is complete and requires only a single power supply (+5 at 1.2 amps) for basic operations and a second supply (+12v at 0.1 amps) for audio cassette operation. The user may elect to operate with either the keyboard and display on the module, or from an external TTY. In either case, he may select addresses, enter data, and start and stop program execution (continuous or single-step). Programs and data may be stored or read from paper tape or audio cassettes.

The KIM-1 system is easily expanded to include additional memory of any variety or additional I/O capability. All necessary bus signals from the microprocessor array are brought to connector pins on the module to simplify the task of expansion.

The KIM-1 system is shipped complete with:

A KIM-1 module (one connector included)
KIM-1 User Manual
KIM-1 System Schematic (Wall Size)
6500 Family Hardware Manual
6500 Family Programming Manual
6500 Programmers Reference Card

Orders currently are being accepted and filled. The system price is $245 (plus $4.50 for shipping and handling). For more information contact MOS Technology, Valley Forge Corporate Center, 950 Rittenhouse Road, Norristown, PA 19401; (215) 666-7950.



CIRCLE NO. 93 ON INQUIRY CARD

## THREE MILLION BIT STORAGE UNIT WITH RS232C WEIGHS ONLY 5.5 LBS.

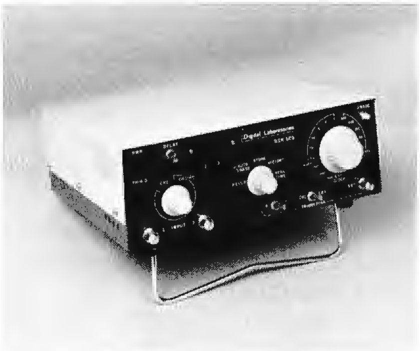
So small that it can be routinely carried in a briefcase or field service kit, the "ACT-1200" is believed to be the world's smallest computer compatible storage unit. Designed primarily for field testing and program loading small computers, it contains a four foot cable and connector that store for travel within the 12.1 × 7.7 × 3.3 inch unit.

The ACT-1200 plugs into any computer or terminal's standard serial asynchronous port, and can record and playback at standard rates up to 1200 baud.

The unit is primarily aimed at the OEM with a population of dedicated minicomputers that have to be occasionally loaded with new programs for upgrading, and the inevitably occuring core "wipe-outs." The ACT-1200 eliminates the need for any on site interfacing or hardware for program loading because the computer's terminal can be unplugged and the ACT-1200 plugged in its place.

Most minicomputers can use existing paper tape oriented software with the ACT-1200. For example, with editing type programs, the ACT-1200 can be used to record source programs following a PUNCH command and re-enter following a READ command. Similarly, the output of an assembler or compiler may be recorded for future loading via standard object loader programs.

Single unit price is $975.00 and delivery is two weeks. The ACT-1200 is manufactured by Digital Laboratories, 377 Putnam Avenue, Cambrige, MA 02139. Phone: 617-879-6220.



CIRCLE NO. 94 ON INQUIRY CARD

## NEW LOGIC ANALYZER MAKES PRICE BREAKTHROUGH

Breaking the $1000 price barrier, the DSR-505 makes available for the first time a low cost unit having all the key features found in this new class of equipment called logic analyzers, digital logic recorders, or logic scopes.

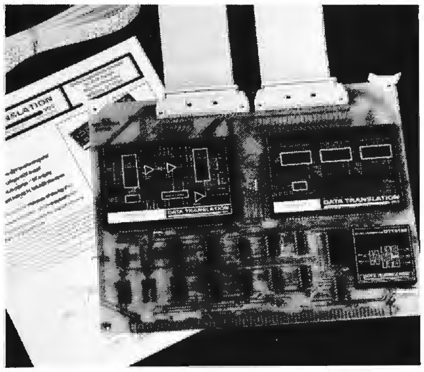
Although the unit is smaller in size and lower in cost than any of the entries so far in this fast growing field, it has several features not found on other instruments. For example, it not only shows the conventional binary highs and lows, but also displays "in between" and open circuit conditions (See Scope Picture). Its controls are unusually easy to use. They function just like those found on a lab scope; in fact, anyone familiar with common storage scopes will be able to operate the unit without instructions.

The DSR-505 monitors its two independent channels continuously, storing each input signal in 512 bit shift registers until a trigger occurs, thus allowing display of signals that occurred prior to the trigger. For example, if an error indicator is used for triggering, then signals that actually lead up to the error may be viewed. This "history" or "pre-triggered" mode provides a completely new tool for finding intermittents and analyzing logic systems.

The DSR-505's time base switch controls the sampling rate so that 10.6 divisions with 48 samples per division on each of two channels are always seen. In normal use, the display scope's controls are not used, with all operations being "referred" to the DSR-505's front panel.

It has 1 megohm, 25 pf. inputs, an internal crystal controlled clock which can be bypassed by an external clock input, a 10 megahertz sampling rate, adjustable thresholds, and pulse stretching that will display a 25 nanosecond pulse on a five second sweep.

Price of the unit is $995.00. Delivery is from stock. The DSR-505 is manufactured by Digital Laboratories, 377 Putnam Ave., Cambridge, Mass. 02139. (617) 876-6220.



CIRCLE NO. 95 ON INQUIRY CARD

## DATA ACQUISITION SYSTEM MATES WITH IMP AND PACE MICROCOMPUTERS

Framingham, MA—Data Translation today announced a completely new data acquisition interface system, designated the DT1722, for the National Semiconductor IMP and PACE series of microcomputers. The new system is designed to fit in one standard slot of the IMP or PACE prototyping system or in the IMP-16C. The DT1722 data acquisition system requires no additional interfacing and plugs directly into the computer. Modules from Data Translation's own DATAX II series are utilized to allow up to 54 analog inputs to be accommodated on the single slot board with power supplied from the computer mainframe. A full 16-channel system sells for $1195.

The DT1722 is the first standardized data acquisition interface board ever offered for the IMP or PACE microcomputers. The single board interface is an ideal low cost solution for acquiring and presenting analog data to microcomputers. It uses standard +5 volt power and is housed within the microcomputer itself, saving the cost of separate power supplies, chassis, and interconnecting cables. Additionally the analog modules are shielded in noise resistant steel cases, preventing EMI/RFI interference in high precision analog circuitry.

The DT1722 offers full 12-bit resolution, accuracy of ±0.03%, either single-ended or differential configuration, and a throughput rate of 25 KHz. Channel addressing is selectable by program control. Device address and analog input ranges are all selectable by jumper plug. Up to 64 channels of analog data may be processed with the single board DT1722.

A unique feature of the DT1722 is its capability of handling a 4 to 20 mA analog input range, a common signal range for industrial systems. This current input range is translated by internal circuitry included on the interface board as are the standard voltage ranges of ±10V, ±5, 0 to 10V, and 0 to 5V.

Data Translation is a leading developer of new minicomputer and microcomputer compatible data acquisition equipment for measurement and control.

For further information contact Fred Molinari, Data Translation Inc., 109 Concord St., Framingham, MA 01701; (617) 879-3595.

# SIMPLIFY YOUR

Boolean Algebra and Karnaugh mapping are two high-powered digital design techniques that will help you design that unobtainable digital circuit easier, faster, and with a little more fun at the effort. Karnaugh mapping is based on the ease with which we recognize visual patterns. Boolean algebra is based on a series of relationships that we will talk about next month. This article gets you into Karnaugh, or K, mapping.

The article started when my wife pointed out a need for a modification to our clock radio. She said that she appreciated the radio when we had to get up to go to work five days out of the week. But she was resentful when it awakened her at the same time on Saturday, rather than allowing an extra hour's sleep before my morning class. She wanted to throw the radio through the wall when it woke us far too early for church on Sunday.

The comment led to a modification that allows us to sleep an extra hour on Saturday and as long as we want to on Sunday. I used K mapping and Boolean algebra in the process. That led to this column with the hope that the techniques will prove useful to you.

Let's get the circuit out of the way first. A TTL 7490 divide-by-ten chip is pulsed every time the clock radio turns on. The output terminals of the 7490 are applied to a gating circuit that connects the speaker to the radio output five out of seven times. The sixth time, the speaker is connected to the radio output after a delay of up to two hours. The seventh time, the speaker is not connected at all. The eight time, the 7490 is reset and the process starts all over again.

Figure one shows the truth table for the output of the 7490 after each input pulse. The line number corresponds to the number of applied pulses. It is also the decimal equivalent of the four bit binary number ABCD.

| LINE | ABCD | S |
|---|---|---|
| 1 | 0000 | 1 |
| 1 | 0001 | 1 |
| 2 | 0010 | 1 |
| 3 | 0011 | 1 |
| 4 | 0100 | 1 |
| 5 | 0101 | 0 |
| 6 | 0110 | 0 |
| 7 | RESET | |

S = ABCD + ABCD + ABCD + ABCD + ABCD

**FIGURE 1**

Figure 1 also shows the expression that defines the operation of the gating circuit that connects the speaker to the output of the amplifier. Output S is true and the speaker is connected immediately if any one of the first five products is true. This circuit could be constructed with 5 four input AND gates and 1 four input OR gate. It would make a lot more sense to simplify the logic expression for S before building the hardware.

A complete four variable truth table has sixteen lines. The corresponding K map has sixteen squares. These squares are arranged as shown in Figure 2a. Figure 2b is the completed K map for the S expression. It was completed by putting a 1 in every square that had a true output on that line in the truth table.

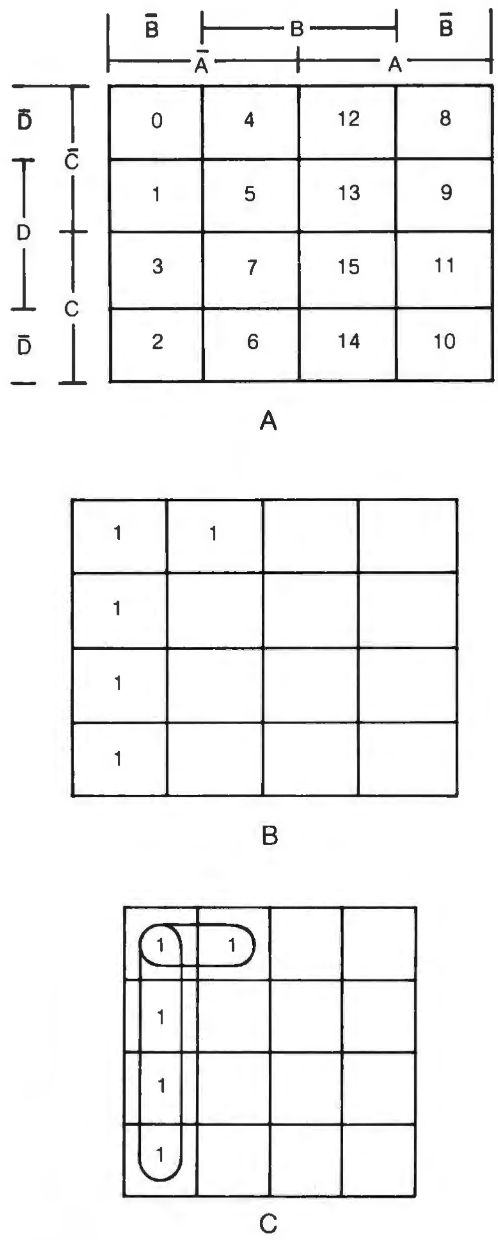



**FIGURE 2**

# DIGITAL DESIGN

*by Bruce A. Scott*

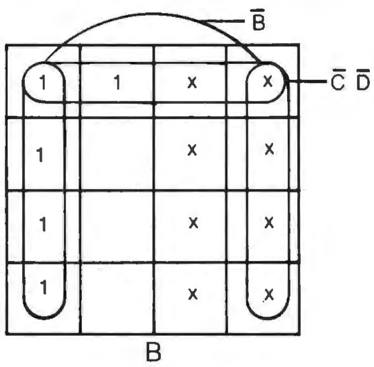
The next step in K mapping is to find groups on the map. There are a few rules that help you do this right. First, the size of the group must be a power of 2, ie: 1, 2,4,8,16 . . . Second, all 1 squares must be covered. Third, each new group must pick up at least one uncovered 1 square. Fourth, all unnecessary groups must be eliminated.

In the example of the clock radio and expression S, squares 0,1,2,3 make up the first group. The second group consists of squares 0 and 4. This is shown in Figure 2c. Notice that square 0 is covered twice.

Reading the groups on the K map is the next step. This is easier if you refer to Figure 2a. Squares 0,1,2,3 are in a column headed with A and B. The other variables appear in both forms, normal and negated, in this group and can be discounted for reasons that we will get into when we talk about Boolean algebra. Squares 0 and 4 comprise a group that belongs to ACD.

The simplification made possible by the grouping reduces the original expression for S to S = AB + ACD. We could construct this circuit with fewer parts and effort than we would need for the first expression for S.

There is a way to make the expression for S even simpler. The original circuit resets the divide-by-ten counter before the counter gets to ten. The input codes for the truth table lines that the counter never sees are totally unimportant. We don't care what the gate circuit does for these inputs. It never sees them. We call these conditions Don't Care and put Xs in the map for these lines of the truth table. Figure 3a shows the K map for S with the Don't Cares added. Grouping can now be done with or without the Xs in any given group. We include the X in the group if the group is made larger. Figure 3b shows the grouping that can now be done using the Don't Care Squares to simplify the expression for S.




This brings up another important point. Notice that the left column is grouped with the far right column. A true K map would be painted on a toroid. If it was, the left border of the map would touch the right border and the top would touch the bottom. That makes other groups possible that are not obvious when the map is shown on a flat sheet of paper.

The final K map expression for S is S = B + CD. We use Boolean algebra to change the form of the expression to S = B (C + D). We can now build this expression into hardware by using one OR gate and one NAND gate a considerable savings over the original expression.

Let me give you two more examples of K mapping to make things a little clearer.

A seven-segment display consists of seven bars. A number is formed by the display when selected bars are lit. For example, the top bar is lit when any of the following numbers are displayed; 0,2,3,5,6,7,8,9. If you used the 7490 divide-by-ten chip to count pulses, you would need a decoding circuit to know when to light the top bar. That circuit would have to satisfy the truth table shown in Figure 4.

| | | | |
|---|---|---|---|
| 1 | 1 | X | X |
| 1 | | X | X |
| 1 | | X | X |
| 1 | | X | X |

A

FIGURE 3

| LINE | ABCD | L | LINE | ABCD | L |
|---|---|---|---|---|---|
| 0 | 0000 | 1 | 8 | 1000 | 1 |
| 1 | 0001 | 0 | 9 | 1001 | 1 |
| 2 | 0010 | 1 | 10 | RESET | X |
| 3 | 0011 | 1 | 11 | | X |
| 4 | 0100 | 0 | 12 | | X |
| 5 | 0101 | 1 | 13 | | X |
| 6 | 0110 | 1 | 14 | | X |
| 7 | 0111 | 1 | 15 | | X |

FIGURE 4.

The unsimplified expression for L is L = ABCD + ABCD + ABCD + ABCD + ABCD + ABCD + ABCD + ABCD. Building this circuit without simplifying L would be a little foolish but it would light the top display bar at the right times.

Let's try a little K mapping to simplify L before we construct the circuit.

The first step is to sketch the K map with sixteen boxes, one for each line of the truth table. The next step is to write a 1 in each box that corresponds to a true output of the gating circuit and an X in every Don't Care box.

These steps have been completed in Figure 5.

| | | | |
|---|---|---|---|
| 1 | | X | 1 |
| | 1 | X | 1 |
| 1 | 1 | X | X |
| 1 | 1 | X | X |

**FIGURE 5**

Now, find the groups. There is more than one way to do this. Perhaps one good way is to find groups that include at least one square that will not fit into any other group. Use the Don't Care Xs where they make things easier. Ignore them where they don't. This effort has been performed in Figure 6.

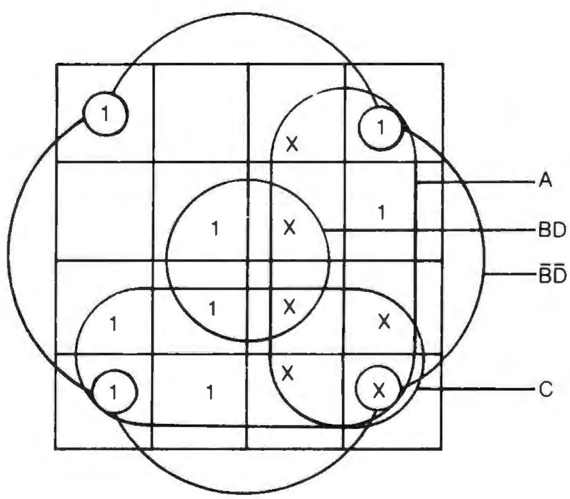



**FIGURE 6**

Finally, read the groups and write them down in a sum of products statement. The simplified expression for L is now L = A + C + BD + BD.

Try a last example that may fool you. You are given the fact that a desired circuit will satisfy the truth table as shown in Figure 7 and you are asked to find a simplified sum of products expression from which to design the hardware.

| LINE | ABCD | F |
|---|---|---|
| 0 | 0000 | 0 |
| 1 | 0001 | 0 |
| 2 | 0010 | 0 |
| 3 | 0011 | 1 |
| 4 | 0100 | 1 |
| 5 | 0101 | 1 |
| 6 | 0110 | 0 |
| 7 | 0111 | 1 |
| 8 | 1000 | 0 |
| 9 | 1001 | 1 |
| 10 | 1010 | 0 |
| 11 | 1011 | 0 |
| 12 | 1100 | 0 |
| 13 | 1101 | 1 |
| 14 | 1110 | 1 |
| 15 | 1111 | 1 |

**FIGURE 7**

Sketching the 16 line K map and filling in the 1 squares produces Figure 8.

| | | | |
|---|---|---|---|
| | 1 | | |
| | 1 | 1 | 1 |
| 1 | 1 | 1 | |
| | | 1 | |

**FIGURE 8**

By this time, you may have noticed that the larger the group is the fewer are the variables that describe it and consequently, the easier it is to build. This might lead to the first choice for a group as the obvious block of four in the center of the map. Wrong.

Falling back on the grouping rules that we used before, we first select those groups that include the squares that cannot be included in other groups. This allows you to complete the K map as shown in Figure 9. The completed K map shows that the large block of four squares forms an unnecessary group. In accordance with our rules, we would eliminate it from the sum of product statement.

**Branch to pg. 60**

While entering a program into a computer, a typing error is usually corrected with ease. A backarrow, or at worst a retyped line will render the mistake erased. This is not true when you are sitting at a typewriter trying to write a letter, or even an article for INTERFACE. If you are a devout hunt-and-peck artist like I am, the odds are that more mistakes will be made than not. And, if you want to have the finished product looking just right, you must also develop the ability to wade through knee-deep crumpled paper.

There are two possible solutions to the problem. One is to scrimp and save and buy a fancy typewriter that has enough brains to correct your mistakes. (There are two types of these. One is delivered in a cardboard box marked "IBM" and the other walks around and looks sexy.) The second possibility is to use your computer as a text editor.

With the program listed below, the user can write text, edit it and print it as many times as needed. For those of you who have both CRT and hard copy terminals, this means that you can write text on a high speed CRT, correct it, and then print it out on your hard copy device.

I decided to write the text editors in Basic for two reasons. First, MITS basic has many nice features, such as string manipulation and input/output control that makes it easy to handle text. Second, like many of you, I don't know how to program in machine language.

**NOTES ON PROGRAM**

I wrote the program on an ALTAIR that was equipped with 12K of memory. This should be enough to write short letters using the text editor if the remark statements are removed. The program is designed to write and edit up to 100 lines of text, but due to the memory limitations of my machine, about 20 lines of 40 characters each is about the maximum I can product. To write more than this, the two following changes must be made:

1. The CLEAR statement in line 100 must be changed to allocate enough string space for your program.
2. The FOR-NEXT loops starting on lines 140, 1510, 2004 and 5710 must be changed to the number of lines of text you want.

Below are some additional notes on the program. They should help you follow it better and aid in trying to tailor the program to suit your need. If you make any super changes, please drop me a description of the changes at the address below.

**INSTRUCTIONS FOR TEXT EDITOR**

1. It is not possible to use a regular INPUT statement for the entering of text. The reason for this is that many characters are not acceptable input. (For example, commas and quotes.) Instead, a WAIT is used along with an INP to input the strings one character at a time. This allows any character, with the exception of three editing characters, to be used in the text.
2. In a few places in the program a PRINT " " appears. A bell (CTRL G) should be placed there. It is used to let the user know that the machine is alive and kicking even when nothing should be echoed.

# MITS BASIC TEXT EDITOR

*by Richard S. Edelman*

When first run, the machine will ask "WHAT FUNCTION?" The functions are:

**WRITE**

This allows the user to write text and edit while writing. Only 70 characters are allowed per line. After this an automatic carriage return/line feed is done, and the next line of text is started.

If a mistake is made, two characters are available for editing. A backarrow will delete the last character input and print the present line, then wait for further input. An "@" will delete the entire line.

To end the text, END must be the last line typed. No carriage return is needed.

NOTE: The program echoes back the input and takes a little longer than BASIC's echoing.

A TAB function is included in the program. The TAB positon is asked for at the beginning ot the *write* section. To tab, type a backslash (SHIFT L).

**PRINT**

The print section will print the text as is. It will print 4 blank lines before and after text. If you are writing a letter the following line should be included in the program to keep "WHAT FUNCTION?" from appearing at the end of your letter:

```
1560 FOR T= 1 to 5000 : NEXT T
```

This should give you time to pull the paper out.

**EDIT**

When edit is called, the machine will ask "TYPE OF EDIT?" The types of editing are:

| | |
|---|---|
| LIST | Lists the text with line numbers and position numbers. |
| DELETE | - Deletes a line or group of characters in a line. |
| INSERT | - Inserts line or character into text. |
| CHANGE | - Change one or more characters in text. |
| SEARCH | - Search for a particular phrase in text and print its position. |

After each individual edit is complete, the program will ask for type of function wanted. To continue editing, EDIT must be typed again.

NOTE: While the machine is in edit mode, a lot of dialogue is used to keep you from making more mistakes when you are trying to fix the first ones.

**END**

This will end the program.

Hopefully, this program will help you become a better typist. But if you are a poor typist, you may not want to type in the program. In this case, a paper tape is available for $3.00. You can drop me a note at:

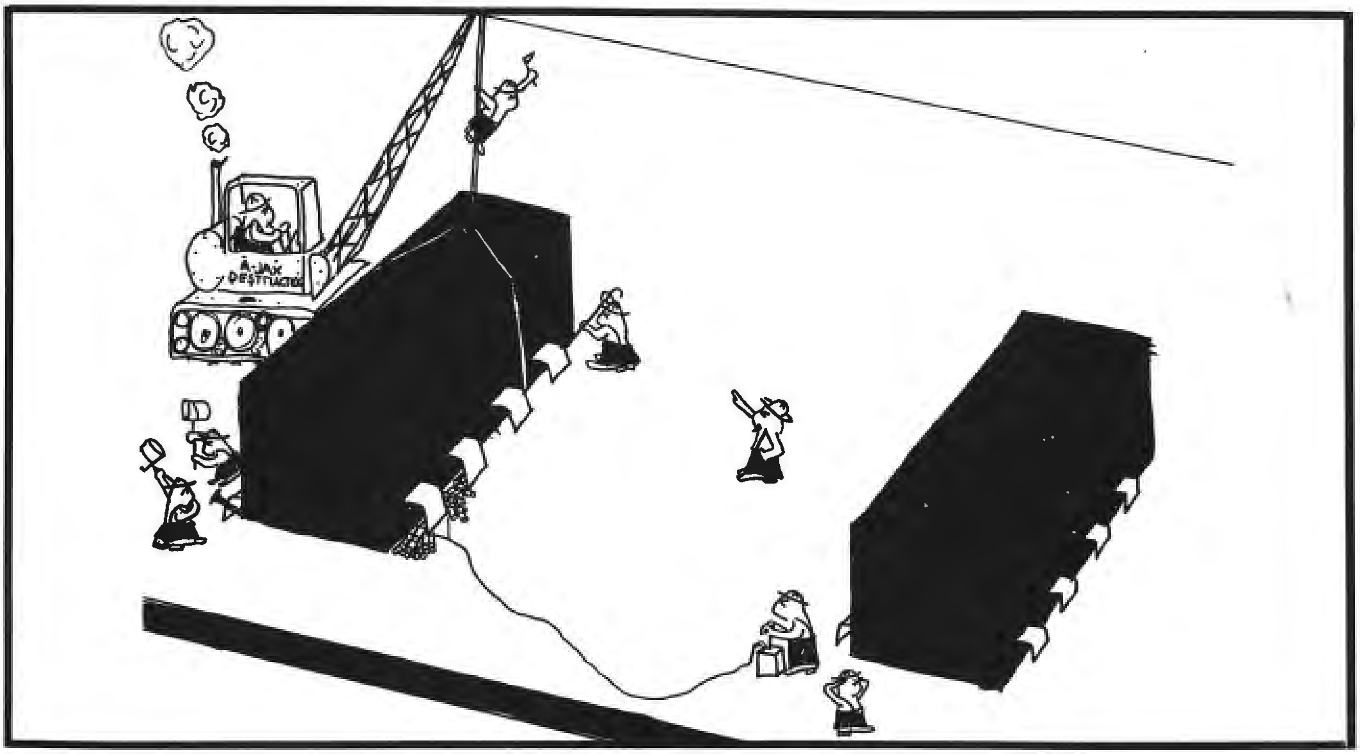
Richard S. Edelman
3626 Goodland Drive
Studio City, CA 91604

```
10 REM **** TEXT EDITOR
20 REM **** FOR ALTAIR BASIC
30 REM **** WRITTEN BY:
40 REM                   RICHARD S. EDELMAN
50 REM                   STUDIO CITY, CALIFORNIA
60 REM                   MARCH, 1976
70 GOTO 300
99 REM *** WRITE SECTION
100 CLEAR 1000
110 DIM A$(100)
111 INPUT"SET TAB AT NUMBER ";T
112 T=T-1
120 PRINT "READY":WAIT 0,1,1
130 PRINT:PRINT:PRINT:PRINT
140 FOR I=1 TO 100
150 K=0
160 WAIT 0,1,1
170 A=INP(1):A=A AND 127
171 IF A=92 THEN 202
180 IF A=13 THEN 280
190 IF A=95 THEN 260
200 IF A=64 THEN A$(I)="":PRINT:GOTO 150
201 GOTO 210
202 B$="":IF K>=TTHEN PRINT"":GOTO 160
203 FOR TAB=1 TO T-K:B$=B$+" ":NEXT TAB
204 PRINTB$;
205 K=K+T-1
207 GOTO 220
210 B$=CHR$(A):PRINT B$;
220 A$(I)=A$(I) + B$:K=K+1
230 IF A$(I)="END" THEN K2=I:GOTO 1500
240 IF LEN(A$(I))>69 THEN 280
250 GOTO 160
260 GOSUB 1000
270 GOTO 160
280 PRINT:NEXT I
299 REM *** PROGRAM CONTROL SECTION
300 INPUT"FUNCTION ";F$
310 IF LEFT$(F$,2)="WR" THEN 100
320 IF LEFT$(F$,2)="PR" THEN 1500
330 IF LEFT$(F$,2)="ED" THEN 2000
340 IF LEFT$(F$,2)="EN" THEN END
400 PRINT "FUNCTIONS ARE:"
410 PRINT "WRITE","PRINT","EDIT","END"
500 GOTO 300
999 REM *** LAST CHARACTER EDIT SECTION
1000 C$=A$(I)
1005 IF K=0 THEN PRINT CHR$(7):RETURN
1010 IF K=1 THEN A$(I)="":PRINT CHR$(7):K=0:RETURN
1020 K=K-1:A$(I)=LEFT$(C$,K)
1030 PRINT:PRINT A$(I);
1040 RETURN
1499 REM *** PRINT SECTION
1500 PRINT:PRINT:PRINT:PRINT
1510 FOR I=1 TO 100
1520 IF A$(I)="END" THEN 1590
1530 PRINT A$(I)
1540 NEXT I
1590 PRINT:PRINT:PRINT:PRINT:GOTO 300
2000 GOTO 2050
```

```
2001 REM *** EDIT LIST SECTION
2002 PRINT:PRINTTAB(5):FOR I=1TO6:PRINT"1234567890";:NEXT
2003 PRINT
2004 FOR I=1 TO 100
2010 IF A$(I)="END" THEN 2050
2020 PRINTI;TAB(5);A$(I)
2030 NEXT I:PRINT:PRINT
2049 REM *** EDIT CONTROL SECTION
2050 INPUT"TYPE OF EDIT ";TE$
2060 E$=LEFT$(TE$,2)
2070 IF E$<>"CH"ANDE$<>"IN"ANDE$<>"DE"ANDE$<>"LI"ANDE$<>"SE"THEN 2090
2080 GOTO 2120
2090 PRINT"TYPES OF EDITING ARE:"
2100 PRINT "CHANGE","INSERT","DELETE"
2105 PRINT"SEARCH","LIST"
2110 GOTO 2050
2120 IF E$="DE" THEN 5000
2125 IF E$="SE"THEN 5600
2130 IF E$="IN" THEN 4000
2134 IF E$="LI"THEN 2002
2139 REM *** EDIT CHANGE SECTION
2140 INPUT"WHICH LINE ";LN
2150 PRINT A$(LN)
2160 INPUT"CORRECT LINE ";CL$
2170 IF LEFT$(CL$,1)<>"Y" THEN 2140
2180 GOTO 3000
3000 INPUT "WHICH CHARACTER ";C
3010 PRINT MID$(A$(LN),C,1)
3020 INPUT "CORRECT CHARACTER ";CC$
3030 IF LEFT$(CC$,1)<>"Y" THEN 3000
3040 INPUT "CHANGE HOW MANY ";N
3050 IF N<0 OR (N+C)>LEN(A$(LN)) THEN 3040
3060 T1$=LEFT$(A$(LN),(C-1))
3070 T=LEN(A$(LN))-(C-1)-N
3080 T2$=RIGHT$(A$(LN),T)
3090 T$="":PRINTT1$;
3100 FOR I2=1 TO N
3110 WAIT 0,1,1
3120 Q=INP(1)
3130 Q=Q AND 127
3140 T$=T$+CHR$(Q):PRINT CHR$(Q);
3150 NEXT I2
3160 PRINT T2$
3170 A$(LN)=T1$+T$+T2$
3180 GOTO 300
3999 REM *** EDIT INSERT SECTION
4000 INPUT "LETTER OR LINE ";LL$
4010 IF LEFT$(LL$,2)="LE" THEN 4050
4020 IF LEFT$(LL$,2)="LI" THEN 4400
4030 GOTO 4000
4049 REM *** LETTER INSERT SECTION
4050 INPUT"WHICH LINE ";LN
4060 PRINT A$(LN):INPUT"CORRECT LINE ";CL$
4070 IF LEFT$(CL$,1)<>"Y" THEN 4050
4100 INPUT "AFTER WHICH CHARACTER ";LE
4110 IF LE>LEN(A$(LN)) THEN 4100
4111 IF LE=0 THEN PRINT"INSERT STARTS AT FIRST POSITION":T1$="":GOTO4143
4120 PRINT MID$(A$(LN),LE,1):INPUT"CORRECT CHARACTER ";CC$
4130 IF LEFT$(CC$,1)<>"Y" THEN 4100
4140 T1$=LEFT$(A$(LN),LE)
```

```
4143 T2$=MID$(A$(LN),(LE+1))
4145 T$=""
4150 PRINT T1$;
4160 WAIT 0,1,1
4170 A=INP(1):A=A AND 127
4180 IF A=13 THEN 4210
4190 T$=T$+CHR$(A):PRINT CHR$(A);
4200 IF LEN(A$(LN))+LEN(T$)=69 THEN PRINT"";
4205 GOTO 4160
4210 A$(LN)=T1$+T$+T2$:PRINTT2$
4220 GOTO 300
4399 REM *** LINE INSERT SECTION
4400 INPUT"AFTER WHICH LINE ";LN
4410 PRINT A$(LN):INPUT"CORRECT LINE ";CL$
4420 IF LEFT$(CL$,1)<>"Y" THEN 4400
4430 LN=LN+1
4440 FOR L=K2 TO LN STEP -1
4450 A$(L+1)=A$(L)
4460 NEXT L:K2=K2+1
4470 PRINT"READY":A$(LN)="":K=0
4480 WAIT 0,1,1
4490 A=INP(1):A=A AND 127
4500 IF A=13 THEN 4560
4510 IF A=95 THEN 4800
4520 IF A=64 THEN A$(LN)="":PRINT:GOTO 4480
4530 B$=CHR$(A):PRINTB$;:K=K+1
4540 A$(LN)=A$(LN)+B$
4550 GOTO 4480
4560 PRINT:GOTO 300
4800 IF K=0 THEN PRINT"":GOTO 4480
4802 IF K=1 THEN K=0:A$(LN)="":PRINT"":GOTO 4480
4803 A$(LN)=LEFT$(A$(LN),K-1):K=K-1
4810 PRINT:PRINTA$(LN);
4820 GOTO 4480
4999 REM *** DELETE SECTION
5000 INPUT"CHARACTER OR LINE ";CL$
5010 IF LEFT$(CL$,2)="CH" THEN 5100
5020 IF LEFT$(CL$,2)="LI" THEN 5400
5030 GOTO 5000
5099 REM *** CHARACTER DELETE SECTION
5100 INPUT"WHICH LINE ";LN
5110 PRINT A$(LN):INPUT"CORRECT LINE ";CL$
5120 IF LEFT$(CL$,1)<>"Y" THEN 5100
5130 INPUT"AFTER WHICH CHARACTER ";N
5140 IF N=0 THEN PRINT"DELETE STARTS WITH CHARACTER 1":GOTO 5170
5145 PRINT MID$(A$(LN),N,1)
5150 INPUT"CORRECT CHARACTER ";CC$
5160 IF LEFT$(CC$,1)<>"Y" THEN 5130
5170 INPUT"DELETE HOW MANY ";N2
5180 IF LEN(A$(LN))-N<N2 THEN 5170
5185 IF N=0 THEN T1$="":GOTO 5200
5190 T1$=LEFT$(A$(LN),N)
5200 T=LEN(A$(LN))-(N+N2)
5210 T2$=RIGHT$(A$(LN),T)
5220 T$=MID$(A$(LN),(N+1),N2)
5230 PRINTT1$;"/";T$;"/";T2$
5240 INPUT "CORRECT ";C$
5250 IF LEFT$(C$,1)<>"Y" THEN 5170
5260 A$(LN)=T1$+T2$
5270 PRINTA$(LN)
```

```
5280 GOTO 300
5399 REM *** LINE DELETE SECTION
5400 INPUT"DELETE WHICH LINE ";WL
5410 IF WL<0 OR WL >=K2 THEN 5400
5420 PRINTA$(WL):INPUT"CORRECT LINE ";CL$
5430 IF LEFT$(CL$,1)<>"Y" THEN 5400
5440 IF A$(WL)="END" THEN 300
5450 FOR J=WL TO K2-1
5460 A$(J)=A$(J+1)
5470 NEXT J:K2=K2-1
5480 INPUT"NEXT LINE ALSO ";NL$
5490 IF LEFT$(NL$,1)="Y" THEN 5440
5500 GOTO 300
5599 REM *** EDIT SEARCH SECTION
5600 PRINT"WHAT PHRASE ";
5605 A1$="":Y=0
5610 WAIT 0,1,1
5620 A=INP(1):A=A AND 127
5630 IF A=13 THEN PRINT:GOTO 5700
5640 IF A=64 OR A=95 THEN A1$="":GOTO 5600
5650 B$=CHR$(A):PRINT B$;:A1$=A1$+B$
5655 GOTO 5610
5700 L=LEN(A1$)
5710 FOR I=1 TO 100
5720 IF LEN(A$(I))<LTHEN 6000
5730 FOR J=1 TO LEN(A$(I))-1
5740 IF A1$=MID$(A$(I),J,L) THEN 5760
5750 GOTO 5950
5760 IF Y=0 THEN Y=1:GOTO 5770
5765 GOTO 5780
5770 PRINTCHR$(34);A1$;CHR$(34);"  CAN BE FOUND ON:"
5780 PRINT"LINE";I;",POSITION";J;". "
5950 NEXT J
6000 NEXT I
6100 IF Y=0 THEN PRINTCHR$(34);A1$;CHR$(34);" IS NOT IN THE TEXT."
6110 GOTO 300
OK
```



# HARDWARE REPORT

## THE WAR OF HARDWARE AND SOFTWARE

*by S. Wilcox*

There is a cybernetic war raging around us. It is a conflict between hardware design and software programming, a battle that is to be won through order and planning or lost in the chaos of coordination failures.

When that intimate relationship known as ownership exists between a computer and its master, it is imperative to understand the interrelationships in the system's hardware and software.

There are apparently some misconceptions regarding some software that is not compatible with certain hardware configurations. For example, in a recent issue of INTERFACE we presented an application program for a Processor Technology 4K memory board to verify its ability to write, retain, and read data from. It may have given the impression that the program would work only with a P.T. memory. This is of course wrong. Any memory board that is, in this case, fully ALTAIR compatible could utilize the program. The author of the program apparently had a P.T. board for which the program was written. However, there is nothing about the hardware that dictates special software. Any of the current ALTAIR/IMSAI memory boards could be tested by this program.

There are instances where software is hardware dependent. It is customary for the minimum hardware configuration necessary for the support of a given piece of software to be called out in the program's documentation. Obvious dependencies are the I/O port addresses that are hardwired to decoder logic, the handshake status/control lines that are dedicated to specific bit positions, and of course the amount of memory occupied by the program or accessible to it.

Input/output format is hardware dependent. It is fruitless to use an I.B.M. Selectric terminal with a program that expects an ASCII encoded device to be connected to the port address that it uses. Similarly a 5 bit BAUDOT coded machine must use a conversion program to transulate the ASCII usually used by a program for I/O communication.

There are other problem areas that can be more subtle. Software critical timing loops are, we feel, generally not desirable to allow favorable exchanges of programs. Arbitrary timing loops necessary to waste some small amount of time are OK, but if a system must generate or use a specific or otherwise accurate time delay, say 0.1385 seconds, trouble can result. Accurate timing is best done by external hardware. But, you say, the CPU clock is crystal controlled, what could be more accurate! Remember software generated timing is dependent on CPU speed which may differ even in the same general family. An 8080 type MPU can run at clock speeds of 2.0 MHz for a plain vanilla 8080 to 4.0 MHz for a 9080A-4. Also memory speed affects timing loops, since wait cycles inserted from slow memory are effectively added to CPU instruction cycle time. An example of a program using critical timing loops is one that performs parallel to serial conversion through software alone.

Interrupt is a great feature allowing more flexibility and speed in running a program that *expects* to see it. You must know if a program is set up to service inputs and outputs via interrupt. You must also know whether or not you must provide hardware to handle and remember multiple nested interrupt requests. A program is essentially blind to system hardware and "discovers" the incompatibility after it bombs.

There are no magic formulas or instant analysis available to say why a particular system fails to operate. There is no substitute for knowledge and logical analysis to determine if hardware has outrightly failed, the software is incorrect, or whether there is a basic incompatibility between good hardware and software that are simply "foreign" to each other.

INTERFACE is interested in hearing from our members and readers about any interesting problems or system anomalies that have been encountered. Our avowed purpose is to exchange information through this medium, with the best data interface being full duplex, so let us hear from you!

## HARD SOFTWARE

ACHTUNG! ALTAIR speed freaks. There is a new product around that can turn your microprocessor based computer into a closer emulation of a minicomputer. GNAT Computers of San Diego, CA has a hardware multiply/divide module now available as a plug-in option for the ALTAIR 8800, the INTEL MDS, the Intellec 8/MOD80, and the GNAT 8080 System. The 8005 type module multiplies and divides in hardware. This type of arithmetic implementation greatly reduces the time required to perform these operations. Software multiply or divide routines can take between 250-400 microseconds. The hard arithmetic approach takes only 2.5 microseconds. These fast operating speeds can be very helpful in floating point, fast Fourier transforms, and other mathematical operations.

The module occupies four memory locations (only one card slot though). Operation is straightforward; the result being produced in two memory locations after the multiplier or dividend is written into one location and the multiplicand or divisor is placed in another. Multiplication of two 8-bit words gives a 16-bit result. Division of a 16-bit word by an 8-bit word gives a 16-bit result with an 8 bit remainder. Division by zero will give FFFF as the result and the low order word of the dividend as the remainder.

The board's operation is so fast that the multiplication and division can be completed before a standard speed 8080 can access the result. There is also a result ready line available as an output to allow use with faster CPU's. The data input/output is buffered with low power (200 microamp) inputs and high power

Branch to pg. 62

## UP 'N RUNNIN'

*by Warren Welmer, Electronics Engineer, Northrup Electronics*

### THE SAGA OF SPHERE

After months of seeing MITS Altair ads, reading about Altair, hearing almost everybody at club meetings talk about Altair, and finally the Altair convention, I decided to write this article about my experiences with the "Avis" (we try harder) of hobbyist microcomputers. I'm talking about that product with the CRT up front, instead of the blinking lights and finger numbing switches of Altair. I'm talking about Sphere, out of where? Utah. Is that near Salt Lake City?

Seriously, let me discuss some of the reasoning for my final choice of a Sphere II System. My electronic engineering background includes an MSEE in computer logic design and thirteen years in industry in the field of testing small airborne digital computers and associated peripherals. I am also a charter member of the Amateur Computer Society (10 years ago) and have been playing with computers for most of those years. My most recent computer system that I have working is a Royal Precision Corp., RPC-4000 with 8K, 32 bit of drum memory, quite slow (250 kHz clock), but not a bad system. Obviously, with my background, a lot of bias enters in and a rank amateur might ask, "How can this guy judge what's a good amateur kit for me, the novice?" Please try to read the bias out.

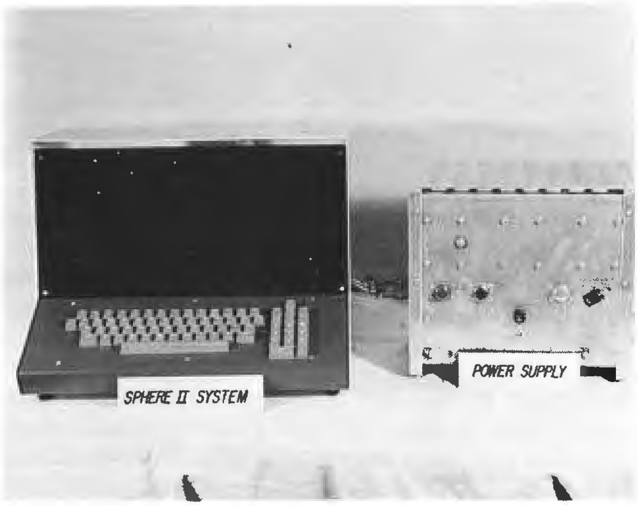
First of all, I did try to survey the field before I bought. I looked at the three or four micros out (last July); Altair, Sphere, Mark 8, SWTP, etc., and said, "As far as price goes, although Sphere is a little more costly at the beginning, all switch flippers are going to spend the money some day for a TTY or TVT." The Sphere also has an attractive cabinet and CRT display. Sphere also sounded like a company that would be around for a while. Not all companies are going to be around when this microprocessor madness is over.

So, with the cash rounded up and the urge to be different, I dropped my order in the mail in August and said, "Well, let's see what happens." You've all had the feeling! About the end of September a box arrives from "where" in Utah! But, it's only a couple of circuit cards, a power supply, and it is labeled Carton 2. Where's Carton 1, where's the CRT monitor and cabinet? It must have gotten lost in the mail. Well, I'll wait since it looks like I have a lot of goodies to play with.

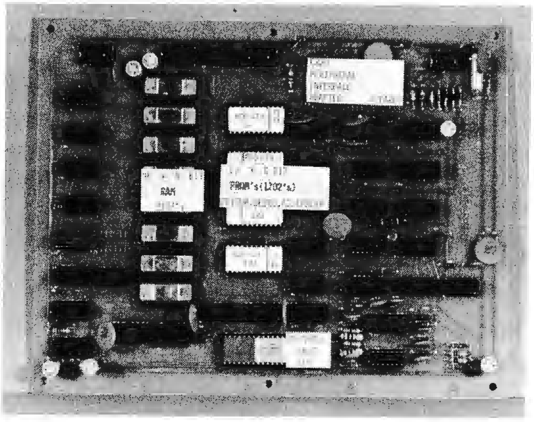
For starters, I thought, "I'd better read the manual; everybody reads the manual first; don't want any goofs." The manual starts out simple; history of computers, good, binary arithmetic, AND gates, computer hardware, software, the Motorola M6800 processor chip; all good. Where's the assembly instructions? I put a Heathkit oscillator together a few years ago and I'm no dummy, but I've got to have some assembly instructions. What is this? I see two or three pages of assembly instructions, a parts list, an old blue line parts layout diagram, and a schematic hot off the designer's desk. Sure enough, the dates on the schematics were July and August dates, but this is only October! I thought, "What have I gotten into; I may be helping design this machine."

With this marginal documentation (Sphere's greatest weakness), I started out on the CPU board. I first took all circuit boards to work for an X-ray so I would have an etch trace diagram for later use if needed. Sphere makes board assembly easy in that all pin ones on integrated circuits face left and all parts are layed out, identified, and stuffed into a foam holding package. The assembly time on the CPU board was approximately four hours. Sphere has a little problem in kitting the proper parts. I wound up short a couple of 10 ohm resistors on the CPU board.

The CRT display board and the keyboard circuit board each took another four hours. I found the CRT board to be kitted with two resistors with incorrect value, a 7409 TTL circuit substituted for a 74123, and a DM 8833 TTL circuit substituted for a 74123. The mistakes were obvious and I caught them before the MIC's were installed. I found it easier to go out to the local electronic parts store to pick up the necessary parts. Sphere charges $5.00 handling fee for any return of parts for replacement whether in warranty or not. The power supply required another eight hours of assembly and was most interesting. The parts layout wasn't bad but from then on it was all "free form art". No wire list was included so you wire from the schematic. The power supply is a separate unit from the CPU and Monitor unit.



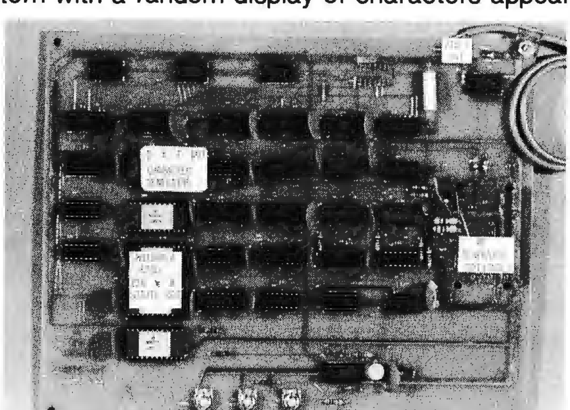

*Sphere II System*



*CPU Circuit Board*

Well, with everything assembled, I was ready for checkout (two weeks later). I take my time. I fired up the power supply, no smoke, so far so good. I checked the voltages, and carefully connected them to the other circuit boards. Sphere uses a bus cabling system instead of mother boards. By this time Carton 1 with the cabinet and video monitor had arrived (no assembly required).

I hooked the CRT display board's composite video output to the monitor's "video in". I fired the whole thing up and "fuzzy trash" appeared on the screen, wow! After a few sync adjustments on the CRT board and monitor, I stabilized the pattern. One control (margin) had no effect. After a little hunting on the associated one shot multivibrator, I found pins 15 and 16 of the MIC shorted by an extra "dogbone" etch between angled conductors going to these pins. Presto! A stable pattern with a random display of characters appeared.



*CRT Display Board*

I hit the RESET and two other keys, SHIFT and CRTL per the manual, but nothing. I decided to take a look at the clock on the CPU board; don't we all start there! It was kind of jittery so I called Sphere. They said remove diode D10 (a clock speed up circuit when not writing in memory).

While I had the scope out, I looked at the clocking and logic on the keyboard, an interesting design but kind of noisy. I'm used to ground planes and nice looking signals in military type computers. The signal out of the 16 to 1 data selector on the keyboard matrix looked awful noisy. I again called Sphere. They suggested a capacitor on this line (.01 uf). The old "hang a cap on it" fix. I've used it myself, shameful from good engineering practice, but often effective.

I tried the RESET again, still nothing. Then, all of a sudden, everything quits and the "old burned resistor" smell permeates the air, a skunk could take lessons, you've all had it happen! All paths lead to the power supply where a .1 ohm one watt resistor in the +5VDC in supply had let go. A quick circuit analysis shows this resistor unrated (needs 2 watts), so I put in a 5 watt resistor. While I was at it, I checked the +5VDC in the CPU/Monitor Unit and found it a little low, about 4.75V. This is the minimum voltage on some TTL MIC's. Since the power supply and the CPU/Monitor are separated by 6 feet of cable (24 gauge wire), I decided to put 12 gauge wire cable between the two units. This helped but I wasn't satisfied. I put an adjustable voltage output circuit in the power supply on the National LM309K (see their catalog) +5VDC regulator. But wait a minute, this sounds like I'm doing design work; oh well, whatever, to get it going. I now had a solid +5.0VDC in the CPU/Monitor unit.

After changing a few keyboard keys (another Sphere weakness), I fired the system up again. With the discovery (after a call to Sphere) that you must push "CTRL" and "E" (for editor mode) keys on the keyboard, I had an operational system, except for the "debug" and "mini-assembler' modes.

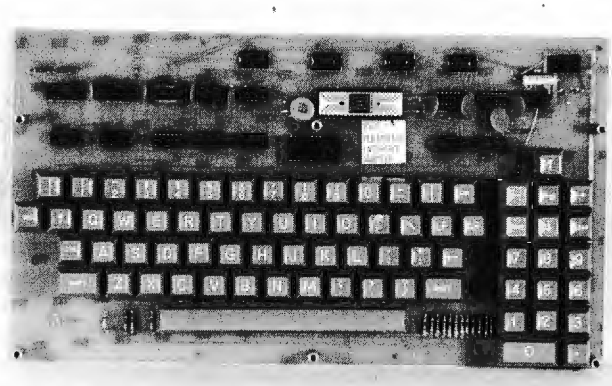
A quick trip to Ernie Dixon's place (Los Angeles representative for Sphere) and some MIC swapping got the "debug" going. It turned out to be a bad RAM in the 4K memory. Sphere uses a portion (200 words) for scratch pad memory for the Editor, Debug and Mini-Assembler program PROM's in the CPU.

After playing with the "mini-assembler" for a while and discovering that blank columns were required in the program listing, I got the sample problem in the manual to run. What a thrill!

I have felt like a pioneer in some aspects of the project (60 Hr of checkout), but have certainly had fun. I think many of the problems I have encountered have been corrected with the updated manual, newsletter, etc.

As estimate to put a kit together now would probably be 25 hours of assembly and 20 hours of checkout. I

**Branch to pg. 62**



*Keyboard Circuit Board*

*Photos by E. J. Smith*

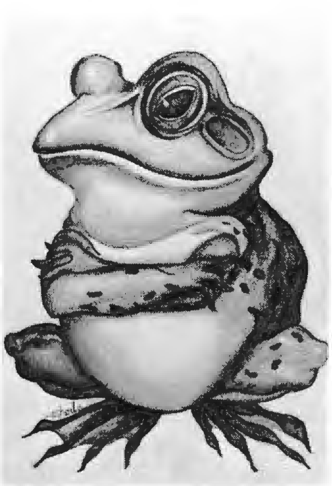


*Photo by Sheila Clarke*

# GAMES & THINGS

*by Tom Rugg and Phil Feldman*

The subject of his month's column is the incredible (?) game of FROG. A game of luck and skill, Frog is adapted from the game of Skunk, and is similar to Pig.

Frog matches a human player against the computer. A pair of dice is used. The only difference between a pair of Frog dice and a regular pair of dice is that each Frog die has a picture of a Frog where the one-spot normally is. The computer simulates the actual dice throwing.

## FROG RULES

The object of the game is to reach the winning point total before your opponent. On a player's turn, he has to roll the dice at least once. If he rolls two Frogs, he loses all points he has accumulated in the game and passes the dice to his opponent. If he throws only one Frog, he loses all points he has accumulated on his current *series* of throws and passes the dice to his opponent. If he rolls no Frogs, he adds the count on the dice to his previous total for this series of rolls and chooses whether he wants to roll again. If he decides not to roll again, he adds the total for his current series to his previous total and passes the dice. If he rolls again, of course, he risks wiping out his score for the series by throwing a Frog.

At the start of each game, the human player is asked what winning score he wants to play up to. Somewhere between 50 and 100 is generally best. You don't automatically win by reaching the winning score, by the way. You have to indicate that you don't want to roll again. Then you are the winner if your score is greater than or equal to the winning score.

## THE FROG PROGRAM

The Frog program shown in Figure 1 was written in Expanded Basic for Control Data 6000 series computers. But take heart, Altair users. Only a small subset was used — no arrays, no string manipulation, no obscure syntax. Every effort was made to avoid using features that would be incompatible with other versions of Basic.

Unfortunately, the random number function (RND) is incompatible with that of many other Basic versions (including Altair's). In this Basic, RND with a negative argument computes a new "seed" for the random number generator based on the time of day. RND with *no* argument produces the next random number in the series based on the current seed. Random numbers are in the range from zero to .9999 . . .

If this isn't compatible with your Basic, you'll want to modify statements 105, 500, 510, and 975. In statement 105 you may want to get your seed from the human player.

The computer's playing strategy is in statements 900 through 995. For maximum enjoyment, we recommend playing the game for a while *before* you try to figure out what its strategy is. Then, if you want to try to modify the strategy (dare we say "improve"?), these are the things to keep in mind:

Before returning, you must set C to either zero or 1. Zero says stop rolling; 1 says to roll again. You can use but not modify the following variables: S (the human's score), R (the computer's score before this series), T (the computer's score this series), and F (the score required to win).

So there you have it, Future Frogplayers of America. We welcome comments, criticism, and improvements. Also, we encourage contributions of games from you readers, especially if you have an original game.

## STAR TREK NEWS

In the last issue we mentioned that plans are being made to film a feature length Star Trek movie, including many members of the original cast and crew.

Unfortunately, the latest news is that filming is now rescheduled to begin in September, rather than July. The script is still being developed. Stay tuned for more details.

```
50  REM  THE GAME OF FROG - MARCH, 1976
60  REM    WRITTEN BY PHIL FELDMAN AND TOM RUGG
100 PRINT "RIBBET"
105 Q=RND(-1)
110 PRINT "HOW MUCH TO WIN? (BETWEEN 50 AND 100 IS BEST)"
120 INPUT F
125 IF F<=0 THEN 110
130 PRINT "WHO GOES FIRST? (1=YOU,0=ME)"
140 INPUT Q
150 S=0
160 R=0
170 IF Q=1 THEN 200
180 IF Q=0 THEN 330
190 GO TO 130
200 GOSUB 800
210 T=0
220 GOSUB 500
230 PRINT "YOU ROLL AND GET A";
240 GOSUB 700
250 IF D=3 THEN S=0
260 IF T=0 THEN 330
270 GOSUB 600
280 PRINT "ROLL AGAIN? (1=YES,0=NO)"
290 INPUT Q
295 PRINT
300 IF Q=1 THEN 220
302 IF Q=0 THEN 310
304 GO TO 280
310 S=S+T
320 GOSUB 800
330 IF S>=F THEN 850
340 T=0
350 GOSUB 500
360 PRINT "I ROLL AND GET A";
370 GOSUB 700
380 IF D=3 THEN R=0
390 IF T=0 THEN 200
400 GOSUB 900
410 IF C=1 THEN 350
420 R=R+T
425 PRINT "I'LL STOP WITH THIS"
427 GOSUB 800
430 IF R>=F THEN 850
440 GO TO 210
500 X=INT(6*RND+1)
510 Y=INT(6*RND+1)
520 D=0
530 IF X=1 THEN D=D+1
540 IF Y=1 THEN D=D+2
550 RETURN
600 PRINT "SCOREBOARD:ME-";R,"YOU-";S,"YOUR SCORE THIS SERIES";T
605 PRINT
610 RETURN
700 IF D=1 THEN PRINT "FROG AND A";Y,"DICE CHANGE HANDS"
710 IF D=2 THEN PRINT X;"AND A FROG", "DICE CHANGE HANDS
720 IF D=3 THEN PRINT "FROG AND A FROG"
730 IF D=3 THEN PRINT "      ------ CROAK"
735 IF D=0 THEN PRINT X;"AND A";Y
736 PRINT
740 T=T+X+Y
750 IF D=0 THEN 770
760 T=0
770 RETURN
800 PRINT "SCOREBOARD:  ME-";R,"YOU-";S,F;"NEEDED TO WIN"
805 PRINT
810 RETURN
850 PRINT "WELL, THATS IT"
860 IF R>=F THEN PRINT "SKILL TRIUMPHS AGAIN"
870 IF S>=F THEN PRINT "BOY YOU WERE LUCKY"
872 PRINT
873 PRINT "HOW ABOUT ANOTHER GAME? (1=YES,0=NO)"
874 INPUT Q
875 PRINT
876 IF Q=1 THEN 100
880 STOP
900 V=R+T
905 IF V>=F THEN 980
910 IF (F-S)<10 THEN 990
915 IF R<S THEN 930
920 L=T/25
925 GO TO 975
930 IF V<S THEN 945
935 L=T/30
940 GO TO 975
945 L=T/35
975 IF RND>L THEN 990
980 C=0
985 RETURN
990 C=1
995 RETURN
```

# Bits & Bytes

by Mike Teener

## POLITICS AND INTRIGUE

There's a lot of politics and intrigue in the microprocessor industry — a couple of case histories may interest you:

MOTOROLA vs. MOS TECHNOLOGY
Once upon a time Chuck Peddle, and seven other engineers who are now with MOS Technology, worked for Motorola's semiconductor division. They were highly involved in the 6800 project, particularly the beginning. Then, not too long ago, and pretty much en masse, they joined MOS Technology. Strangely enough, the new MOS Technology 6501 was pin-compatable with Motorola's 6800 . . . even the support chips (PIA) were the same. So Motorola sued and just recently won an out-of-court settlement that has MOS Technology paying $200,000 and stopping production on the 6501. The similar, but better, 6502 will still be around, so don't cry about it.

GENERAL INSTRUMENTS vs. FAIRCHILD GI is suing Fairchild for "misappropriation of trade secrets". It seems GI got some information on a microprocessor design from Olympia Werke AG, a German electronics firm, in a non-exclusive licensing agreement (this means that Olympia could sell information to some other firm if they wanted to). This information, in turn, was transferred to Fairchild via a former GI employee. GI claims the information was used to bring out the F8 microprocessor. Now Fairchild itself is entering into a licensing agreement with Olympia — legally obtaining the information that GI claims Fairchild stole. Reading about this case is fun . . . every paragraph is full of "alleged theft" and "alleged secrets" and "prior consent" and "plaintiff" and "counsel". Fabulous!

Branch to pg. 63

# Book Review

## MACHINE LANGUAGE PROGRAMMING FOR THE 8008

**(AND SIMILAR MICROCOMPUTERS)**
**Scelbi Computer Consulting, Inc.**
**1322 Rear — Boston Post Road**
**Milford, CT 06460Price: $19.95**

**Reviewed by Walter M. White**

A must for the 8008 owner, this book begins with a thorough description of the 8008 instruction set and builds in complexity to the sophistication of complete routines for performing floating point arithmetic. It is ideal for the beginner who has limited knowledge of what is involved in programming a microprocessor.

Written by Nat Wadsworth of the Scelbi staff, most of the chapters can apply to any processor which has an instruction set similar to that of the 8008, such as the 8080.

The book enables the beginning microprocessor owner to build machine language programs without the use of editors, assemblers and other sophisticated tools usually found on larger systems. In the early chapters, flow charts, the octal (base 8) numbering system, and some basic concepts such as memory pointers, counters, and the use of the accumulator form a base on which more intricate programs are built.

Various methods of handling tables and code conversions, character strings and sorting operations are followed by routines to perform floating point addition, subtraction, multiplication and division. In addition to the source program listings of these math routines, a complete assembled listing with 8008 codes is included along with an operating program to demonstrate the routines. It is unfortunate that no overflow or underflow checks are included, although the ambitious reader should have no trouble in modifying the routines to include these checks once the math processes are fully understood.

Binary to decimal and decimal to binary conversion is also covered. One third of the book is devoted to these arithmetic functions which can turn a microprocessor toy into a useful and powerful tool.

The routines as presented in the book were designed to make the logical development of the programs clear to the reader. Therefore, they were not necessarily designed for minimum memory space nor for fastest execution time. Chapter six is devoted to using memory more efficiently, although I disagree with their suggestion that the single byte RESTART instructions be used as general subroutine calls. As the computer hobbyist expands the hardware capabilities of his system, the RESTART instructions will become more useful when interrupt capabilities are added.

The mnemonics used throughout the book are the old standard originally used by Intel where every instruction mnemonic was three characters long (MBA rather than MOV B,A and DCE rather than DCR E). The reader should have no trouble understanding them, however, as they are fully described in chapter one.

In summary, this book would be a good first programming book for the hobby computer enthusiast. (8½ x 11 inches, soft cover, 180 pages.)

## COMPUTER LIB

**Theodor H. Nelson, Publisher**
**P.O. Box 2622**
**Chicago, Ill. 60690**
**Price: $7.00**

**Reviewed by Jon Walden**

*The following book review appeared in Vol. 1, #1, (Sep. '75) of* Interface Newsletter, *edited by Jon Walden. The book COMPUTER LIB, warrants the rerun, and Jon's review is worth reading again.*

I'm delighted to have the chance to sing out about Theodor H. Nelson's book, "Computer Lib," and its flip side, "Dream Machines." I could wax eloquent with superlatives (and I probably will)! Not only about the inviting cameos ("articles" or "chapters" would fail to reflect the grab this book has) which range from the elementary to the profound and cover more aspects of the field than I had ever suspected to exist . . . but also about the *spirit* of this book.

It's the man's attitude, his vision, his *head* that spaces me. He leaves me *high*, in the finest sense of that word. Let me throw some quick quotes at you:

There is no question of whether the computer will remake society; it has.

"Rigid and inhuman" computer systems are the creation of rigid and inhuman people.

Microprocessors are what's happening.

The technicalities matter a lot, but the unifying vision matters more.

Add to this a lot of solid information about microprocessors thru the biggies, languages, operating systems, time-sharing, peripherals, applications (*many* ideas to explore), and you can begin to see why I get so turned on . . .

What I would really like to do is just reprint the whole book for you! No? Then give up a few martinis or a pair of new jeans and shell out the seven bucks. Dick or Lois (Heiser) will sell it to you at the Computer Store or you can order it thru PCC. Whether you're just starting out or have been toying with computers since vacuum tube days, you'll never again wonder what to pick up when you find a few minutes to spare.

Only one caution: Keep a magnifying glass nearby; the data is packed in tight little characters that tend to run together far sooner than you're ready to quit

reading!

Computer Lib? Ted Nelson defines it as "making people freer through computers. That's all."

*Mr. Nelson plans his fourth printing . . . he is his own publisher. We believe that anyone who's interested in anything about home computing isn't really, unless this book is in his (or her) "how to" library.* ▣

**EXCHANGE** vectored from pg. 10 . . .

good place to start with ICs. If you are unable to turn up a copy, send me a self addressed stamped envelope and 25¢ and I'll send you a copy of the article the following day of receipt.

I'd be pleased to learn of other tutorial materials and experiences which the readers have found of value—let me know so I can pass them along. ▣



**SERIAL DATA** vectored from pg. 28 . . .

made. The second area will be of concern only where dynamic RAM memories are used. The memory refresh will also affect program execution times (except in the rare case where refresh is "transparent"). The program illustrated in this article will be affected little by either of these considerations since it is "self-synchronizing"—but other timing applications may require additional investigation.

In this month's article, I have shown many ways that a microcomputer can be used to reduce the amount of hardware in a low cost application. The hardware elements that have been eliminated may not even be readily apparent to the casual observer. I think it is obvious to everyone that we have replaced a UART—but what about the counter chips (program instructions perform timing), filter (debounce and verification of START bit), decoder (linear select), one-shots, flip flops and other logic gates that have been replaced by programmed logic? We have only touched the surface of potential microcomputer application techniques but will be discussing additional topics in future articles.

Next month I will discuss interrupts, showing how they can be used to make the program illustrated in this article more practical for use in serial communication. The effect of interrupts on timing applications will also be demonstrated. ▣

[1]Intel 8080 Microcomputer Systems User's Manual. $5.00. Intel Corporation, 3065 Bowers Ave., Santa Clara, CA 95051.

**DIGITAL DESIGN** vectored from pg. 48 . . .

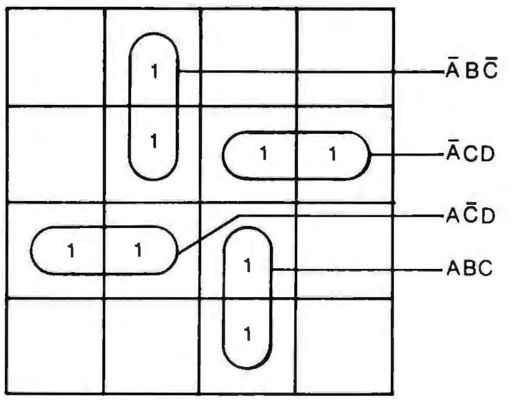



**FIGURE 9**

The final simplification for F is F = ABC + ABC + ACD + ACD.

This completes a brief introduction to Karnaugh mapping. Next month we will take a similar look at the techniques of Boolean algebra and show you how it can make your circuit design work easier and a little more fun. ▣

# Group Purchase For SCCS Members

A group purchase program has been established to enable all SCCS members to participate in the buying of selected products at discounted rates. To find out products and prices, a taped telephone message is prepared monthly, (213) 425-5120, with updated information. The tape states the date of the recording so callers can verify that they have current data. Announcements of available products will be made at each SCCS monthly meeting, and the tape is then updated. But please allow three days after the meeting for preparation of the new message before calling.

Since the message is stated only once and very quickly, it is suggested that a cassette recorder be used to make a permanent record of the information. The message will mention the product name and price only. For complete details on products listed, refer directly to the manufacturer or product vendor.

**ORDER DEADLINE IS THE FIFTH OF EVERY MONTH.** Orders received on the fifth of the month prior to meeting date will be delivered at the meeting. Group purchase is based upon quantity requirement—if the minimum is not met, either a refund of your money, or holdover of delivery until the following month, may be required. Those who wish to receive shipment by UPS must add 5% to their purchase price to cover shipping and insurance. A 2% add-on (or $1, whichever is greater) is required to cover handling costs. California residents must add 6% sales tax to the total purchase price.

Send your check or money order, made out to the Southern California Computer Society Trust A, along with the order form below.

**GROUP PURCHASE ORDER FORM**

Clip and mail this order form, with your check or money order, to

SOUTHERN CALIFORNIA COMPUTER SOCIETY
3238 Faust Street
Long Beach, California 90808

Member Name ______________________________

Membership # ______________________________

Address ______________________________
Street City State Zip

I will pick up my order at the next SCCS meeting ☐

Please ship my order to me, for which I have included 5% for shipping and insurance. ☐

| QUANTITY | ITEM/DESCRIPTION | UNIT PRICE | TOTAL |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | |
|---|---|
| Sub Total | |
| 2% Handling ($1 minimum) | |
| 6% Sales Tax | |
| 5% Shipping and insurance | |
| TOTAL ENCLOSED | |

# AN OUNCE OF PREVENTION

*by Cassandra Cassette*

The year 1976 is the 200th year of the American revolution and the second year of the computer hobbyist revolution. Everything looks great. There are more than a dozen computer kits to choose from. The cost of hardware and software is dropping steadily. Thousands of hobbyists are joyfully running their favorite games, creating their own programming languages, synthesizing music, and designing their own computer systems. The personal computer is indeed the ultimate toy.

But a small computer can also be much more than just a toy. As some hobbyists have already discovered (and microsystems engineers have known all along), small computers can do many things that people are willing to pay money to have done — things like billing, bookkeeping, inventory, payroll, scheduling, process control, and a host of other things traditionally done by dinosaurs with strange names like Brontosaurus 3300, Stegosaurus 1108, and 370 Rex. The saurians don't know it yet but they have been superseded by small furry things. When they finally notice that their numbers are diminishing drastically, they will look around for someone to blame.

Like us.

They may snort and bellow "Unfair competition! We have been underpriced by these impudent upstarts with their 8080's, 6800's, and 1800's operating out of cellars, attics, garages, and gazebos. How dare they do this to us? There ought to be a law!"

Aye, there's the rub. Before sinking into the depths of the tar pits, the saurians could cause a lot of trouble by demanding that computer hobbyists be regulated, taxed, or even outlawed. Don't think it couldn't happen. For instance: Are you zoned for programming?

Not long ago, the news media carried a story about two young students who were doing freelance programming at home in their spare time until someone complained and forced them to get a business license and rent an office. This unexpected expense probably wiped out most of the income from their programming project.

Ask your friendly neighborhood ham radio or CB fan how he likes being regulated by the FCC. He may tell you that it's already illegal to transmit ASCII code via ham radio; only Baudot TTY code is allowed, and that only on certain bands.

The regulatory harassment of the computer hobbyist is potentially unlimited, bringing to mind the following grim scenario:

The time is 1984, the place is Dataville, U.S.A. At the city limits guards are searching all incoming vehicles for contraband such as organic vegetables or IC chips. Around the corner in a dark alley lurks a disreputable looking character (his tattered overcoat concealing his dragon T-shirt) trying to sell a Star Trek tape to a wary juvenile. Down the block, newspaper headlines announce the confiscation of two kilos of floppy disks —the biggest data-bust in months.

Improbable? Let's hope so. Perhaps the best way to forestall onerous regulation is through education. Hobbyists should always respect the rights of hardware patent holders and software copyright owners. Before starting a commercial computing venture it might be wise for the hobbyist-turned-pro to seek expert legal advice on how to go about it. As a last-ditch measure, it could even be necessary for computer buffs to present a united front and send a hobbyist lobbyist to Washington.

Although there can be little doubt that this is the heyday of the happy hacker, it may turn out to be later than we think.

---

**UP 'N' RUNNIN' vectored from pg. 56 . . .**

still think that the Sphere is a good system and with the Cassette interface, Modem, and BASIC software promised shortly, will be one of the most powerful hobbiest systems available.

My usage will initially be for games and fun. Later I hope to work into the data processing business when I can get a line printer and disc system hooked up to the Sphere. So, watch out number one Altair, here comes Sphere!

---

**REPORT vectored from pg. 54 . . .**

(40 milliamp) outputs. The other buss lines are all low power Shottky. Maximum power draw is 2 amps for the 2.3 microsecond module. However, there is also a 5 microsecond option that consumes only 600 ma. Both types require only a standard 5 volt power source.

At the present time this option is available assembled only and tested, although a kit version is planned at a lower price and will be marketed through selected computer hobbiest type outlets. Prices are $225 for the 5 microsecond multiply/divide module, and $275 for the 2.5 microsecond version.

Module part numbers:
8005 for GNAT 8080 System
8006 for ALTAIR 8800
8007 for INTEL MDS
8008 for Intellec 8/MOD80

For further information contact:
GNAT Computers
8869 Balboa, unit C,
San Diego, Calif. 92123 (714) 560-9433

FIREWORKS vectored from pg. 32 ...

invention of fireworks.

Next month, with some help from my programming friends, we will bring to you some notes on writing harmonic algorithms for graphic programs.

BITS & BYTES vectored from pg. 58 ...

NEW 6800 PRICING

Speaking of Motorola, there are new prices for the 6800. $35 for 1-9, $32.50 for 10-49, and $29.95 for 50-99. The great price war continues. Also expect a super new version — twice as fast!

RCA has a new way to buy their COSMAC micro, the CDP1802, a one-chip CMOS 40-pin design. It has all the old CDP1801 features like 3 to 15v. single unregulated power supply, 16-16 bit registers, 64K byte addressing and all that. What's new is low price ($23.50 for 100 and up), high speed (2.5-3.75 microsecond per instruction), fancy DMA capability, and a host of CMOS support chips (including multiply/divide and PIS/ACIA type chips).

NEW APL PROGRAMS PROMISED

I may have to take someone out to lunch fairly soon; Micro-Soft — the people who wrote the very good MITS BASIC — claim to be working on both an 8080 APL and a 6800 APL. They are not very happy about the small amount of money they actually got for writing the MITS BASIC, so you and I may not get their new programs. I certainly hope they do become available — their stuff is really first-rate.

P.S Still no Altair 680. What can I say?

# *Unclassified Advertising.....*

## *Caveat Emptor*

**MEMBERSHIP CLASSIFIED ADVERTISING**

Each month SCCS Interface will devote some space free, non-commercial advertising by members. This is done as a service to members of the Southern California Computer Society and to help promote communications between computer experimenters and hobbyists throughout the world.

Many SCCS members, being pioneers in the small computer field, often have for sale, on an occasional basis, used computing equipment, or equipment they have designed and built themselves. Those members, as well as those who simply wish to establish contact with individuals with similar or complimentary interests are encouraged to utilize the member ad privilege subject to the following conditions:

A. Ads must be received by the ad editor of SCCS Interface by the first of the month prior to issue month

B. Ads may list the member's name, home address and phone. No commercial firm names, please!

C. Ads must be typewritten, double spaced and no more than 250 characters (including spaces and punctuation) in length. Ads 250-500 characters will be printed on space-available basis.

D. Ads may be run up to three consecutive issues, however, a specific request must be received for each instance.

E. No free member ads will be accepted for the sale of equipment or services by a commercial profit-oriented individual or organization.

Address all member ads to: Interface Ad Editor, Nancy DeLong, 8005 Denrock Ave., Los Angeles, Calif. 90045

Those non-members wishing to advertise are invited to join SCCS and enjoy the member ad privilege as well as the delivery of SCCS INTERFACE to your mail box.

*FOR SALE: REDCOR R.C.-70 MIDI COMPUTER KIT, Partially assembled with 65K Bytes of Core Memory, DMA & Interrupt & Selectric Controller Cards. FORTRAN and BASIC Compilers available. Asking $1000. Also have IBM 727 Tape Drives, $150 each F.O.B. Contact Buster Killion, 2773 Winrock, Altadena, Ca. 91001 (213) 798-2977.*

*FOR SALE: ALTAIR 8800 w/2k memory, fan, P.T.Co. 3P+S I/O, moboard, cardcage, 16v transformer, buss terminator. Good working Condx. $850 value, asking $500 for package. Contact John Moorhead, 928 J Street, Davis, Ca. 95616. Phone MTFSSUN eves. (707) 758-2495.*

*FOR SALE OR SWAP: CCTV-Video Tape Recorder (Cartrivision), with solid state b-w camera (zoom lens); microphone and cables; modulator for antenna terminal connection; large number of tapes, including color demo tape. For: UP components (8080), memory, I/O, etc. Or what have you. Cash: $500. (I'll pay shipping costs). Call or write Pat Rankin, 1085 Tasman #29, Svale, Ca. 94086. (408) 734-1614 after 5 p.m. and weekends.*

*NEED HELP!!! with computer car pool for meetings and classes. Also phone network for club communications and notices. Contact Louis G. Fields, Vice President SCCS, 11662 Sunset Blvd., Suite 2, Los Angeles, Ca. 90049. (213) 472-0388 or 272-0942.*

*WHO NEEDS IBM? Are you using a DEC system-10? If you are interested in starting a club solely devoted to the 10, please call Ralph Klestadt at 784-8319 or 789-9616 after 4 p.m. Remember, you are not alone out there with your 10!*

*ALTAIR 8800, fully assembled, 100% functional, all IC's socketed, fan, PTCO full 16 slot mother board, no memory, documentation; $450. J. Ellmers, 841 Kinderkamack Rd., Oradell, NJ 07649.*

*Handsome engraved membership badges for members, officials and officers. $3.50. Help people know who you are. Contact Louis G. Fields, membership Chairman, 11662 Sunset Blvd., Suite 2, Los Angeles, Ca. 90049. Call (213) 472-0388 or (213) 272-0942.*

*HAMS & CB'RS Please contact SCCS VP Louis G. Fields, 11662 Sunset Blvd., Suite 2, Los Angeles 90049 (213) 472-0388 or (213) 272-0942.*

*Save my marriage! Buy my new assembled IMSAI 8080, Loaded 22 slot mother board, 8k Ram, regular price, $1835.00. Will sell to highest bidder above $1700.00. Also, IMSAI 8080 kit, still in box, large mother board, regular price $587.00. Will sell to highest bidder above $547.00. Send bids to 664 Via Alamo/San Lorenzo, Ca. 94580*

*MICROSYSTEM CONSULTANT for your special requirements for systems or hardware design. Ruben Loshak, 4040 Badillo Circle, Apt. 50, Baldwin Park, Calif. 91706. (213) 338-8549.*

*Bill Fornaciari says, "I am in a wheelchair, and I would be interested in pooling a ride to some of the meetings: I can offer vehicle." Contact Bill at (213) 449-8928 or write 1070 Arden Rd., Pasadena, 91106.*

# KUPON KORNER

**ALTERNATE COUPON**
**AS SEEN IN SCCS INTERFACE**

Enclosed is a check for $________
BankAmericard # ________ Expiration Date________
Master Charge # ________ Expiration Date________
ITEM 1________ Price________
ITEM 2________ Price________
ITEM 3________ Price________
ITEM 4________ Price________
Include Postage Total________
Name________
Address________
City ________ State & Zip________
Send Directly to Advertiser Address

**ALTERNATE COUPON**
**AS SEEN IN SCCS INTERFACE**

Enclosed is a check for $________
BankAmericard # ________ Expiration Date________
Master Charge # ________ Expiration Date________
ITEM 1________ Price________
ITEM 2________ Price________
ITEM 3________ Price________
ITEM 4________ Price________
Include Postage Total________
Name________
Address________
City ________ State & Zip________
Send Directly to Advertiser Address

## ADVERTISING INDEX



## NEW PRODUCT INDEX



# Classified Advertising

# MEMBERSHIP APPLICATION

**MEMBERSHIP/SUBSCRIPTION APPLICATION FORM**
SOUTHERN CALIFORNIA COMPUTER SOCIETY
P. O. BOX 3123
LOS ANGELES, CALIFORNIA 90051
**PAYMENT MUST ACCOMPANY FORM**

DUES: $10.00/year ($1.00 included for publication subscription)
Make check payable to: SOUTHERN CALIFORNIA COMPUTER SOCIETY
**ADDRESS SHOULD READ: (please type or print)**

Name__________ Title __________

Company __________

Home Address __________

City________ State ________ Country ________ Zip ________

Date________ Signature ________


# MEMBERSHIP APPLICATION

**MEMBERSHIP/SUBSCRIPTION APPLICATION FORM**
SOUTHERN CALIFORNIA COMPUTER SOCIETY
P. O. BOX 3123
LOS ANGELES, CALIFORNIA 90051
**PAYMENT MUST ACCOMPANY FORM**

DUES: $10.00/year ($1.00 included for publication subscription)
Make check payable to: SOUTHERN CALIFORNIA COMPUTER SOCIETY
**ADDRESS SHOULD READ: (please type or print)**

Name__________ Title __________

Company __________

Home Address __________

City________ State ________ Country ________ Zip ________

Date________ Signature ________


## READER SERVICE CARD

## SCCS INTERFACE

**May 1976 Issue — Void after July 31, 1976**

**(Please type or print)**

Name__________ Title __________

Company__________ Telephone (A/C)__________

Home Address __________

City________ State________ Country________ Zip________

Please send information on items circles below.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |
| 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 |
| 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 |
| 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 |
| 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 |

What products and/or services are you planning to buy and/or specify within the next 90 days?

____________________


## READER SERVICE CARD

## SCCS INTERFACE

**May 1976 Issue — Void after July 31, 1976**

**(Please type or print)**

Name__________ Title __________

Company__________ Telephone (A/C)__________

Home Address __________

City________ State________ Country________ Zip________

Please send information on items circles below.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |
| 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 |
| 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 |
| 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 |
| 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 |

What products and/or services are you planning to buy and/or specify within the next 90 days?

____________________

## READER SERVICE CARD

## SCCS INTERFACE

**May 1976 Issue — Void after July 31, 1976**

**(Please type or print)**

Name__________ Title __________

Company__________ Telephone (A/C)__________

Home Address __________

City________ State________ Country________ Zip________

Please send information on items circles below.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |
| 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 |
| 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 |
| 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 |
| 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 |

What products and/or services are you planning to buy and/or specify within the next 90 days?

____________________

FIRST CLASS
PERMIT No. 209
Cerritos, Ca. 90701

**BUSINESS REPLY MAIL**
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Interface Hotline
P.O. Box 1234
Cerritos, Ca. 90701

# PLANNING TO MOVE?

Let us know 8 weeks in advance so that you won't miss a single issue of SCCS INTERFACE.

Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly.

**If you have no label handy, print OLD address here.**

Please Print

Name

Company

Address

City State Zip code

**Affix Label** **Print NEW address here**

Name

Company

Address

City State Zip code

**MAIL COMPLETED CARD TO:**
**Southern California Computer Society**
P. O. Box 3123
Los Angeles, California 90051

FIRST CLASS
PERMIT No. 209
Cerritos, Ca. 90701

**BUSINESS REPLY MAIL**
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Interface Hotline
P.O. Box 1234
Cerritos, Ca. 90701

# PLANNING TO MOVE?

Let us know 8 weeks in advance so that you won't miss a single issue of SCCS INTERFACE.

Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly.

**If you have no label handy, print OLD address here.**

Please Print

Name

Company

Address

City State Zip code

**Affix Label** **Print NEW address here**

Name

Company

Address

City State Zip code

**MAIL COMPLETED CARD TO:**
**Southern California Computer Society**
P. O. Box 3123
Los Angeles, California 90051

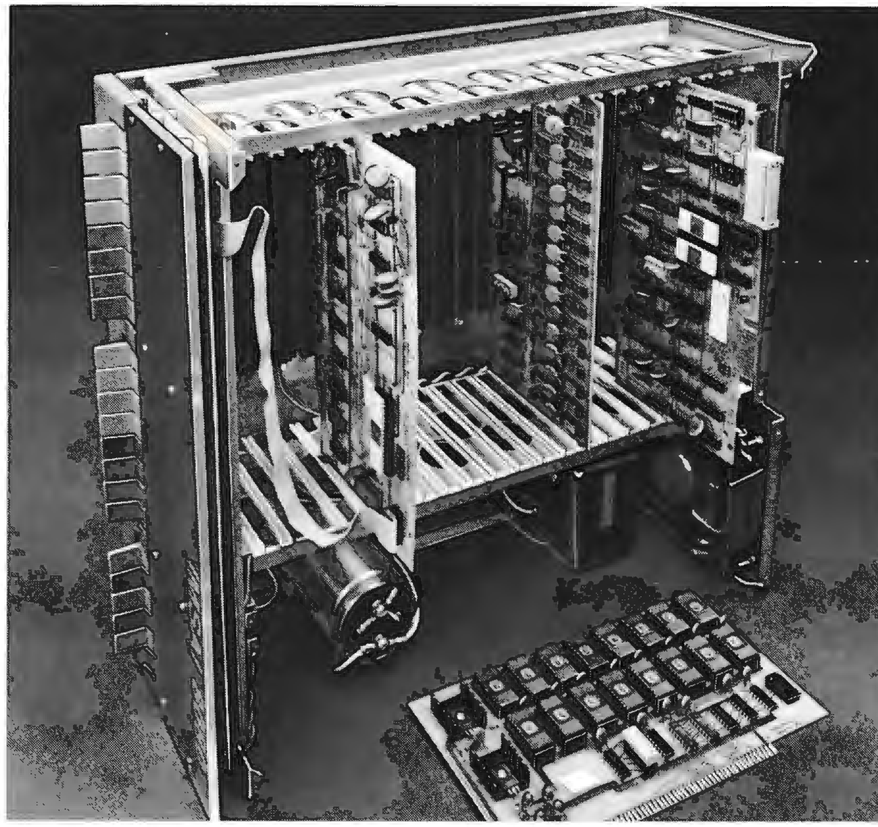
FIRST CLASS
PERMIT No. 209
Cerritos, Ca. 90701

**BUSINESS REPLY MAIL**
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Interface Hotline
P.O. Box 1234
Cerritos, Ca. 90701

# If you thought a rugged, professional yet affordable computer didn't exist,

# think IMSAI 8080.



Sure there are other commercial, high-quality computers that can perform like the 8080. But their prices are 5 times as high. There is a rugged, reliable, industrial computer, with high commercial-type performance. And prices that are competitive with Altair's hobbyist kit. The IMSAI 8080. Fully assembled, it's $931. Unassembled, it's $599. And ours is available now.

In our case, you can tell a computer by its cabinet. The IMSAI 8080 is made for commercial users. And it looks it. Inside and out! The cabinet is attractive, heavy-gauge aluminum. The heavy-duty lucite front panel has an extra 8 program controlled LED's. It plugs directly into the Mother Board without a wire harness. And rugged commercial grade paddle switches that are backed up by reliable debouncing circuits. But higher aesthetics on the outside is only the beginning. The guts of the IMSAI 8080 is where its true beauty lies.

The 8080 is optionally expandable to a substantial system with 22 card slots in a single printed circuit board. And the durable card cage is made of commercial-grade anodized aluminum. The Altair kit only provides 16 slots maximum in four separate sections, each section requiring 200 solder connections.

The IMSAI 8080 power supply produces a true 28 amp current, enough to power a full system. The Altair produces only 8 amps.

You can expand to a powerful system with 64K of memory, plus a floppy disk controller, with its own on board 8080—and a DOS. An audio tape cassette input device, a printer, plus a video terminal and a teleprinter. These peripherals will function with an 8-level priority interrupt system. IMSAI BASIC software is available in 4K, 8K and 12K, that you can get in PROM. And a new $139 4K RAM board with software memory protect.


Find out more about the computer you thought didn't exist. Get a complete illustrated brochure describing the IMSAI 8080, options, peripherals, software, prices and specifications. Send one dollar to cover handling to IMS. The IMSAI 8080. From the same technology that developed the HYPERCUBE Computer architecture and Intelligent Disk systems.

Dealer inquiries invited.

IMS

**IMS Associates, Inc.**
14860 Wicks Boulevard
San Leandro, CA 94577
(415) 483-2093

CIRCLE NO. 16 ON INQUIRY CARD

# SPHERE SURVEY DEFINES YOUR COMPUTER NEEDS

After talking to thousands of potential and existing computer users like you, we have found that what you want is a computer that has these features:

- **A KEYBOARD** to input your programs.
- **A PROCESSOR** with software that lets you program immediately with power-on.
- A **CRT** to **DISPLAY** your programs so you can see what you are doing.
- MEMORY for rapid access of **STORED DATA** and **PROGRAM OPERATIONS.**
- AN **AUDIO CASSETTE** to save for **FUTURE** access all of the data you have generated.
- **INPUT/OUTPUT** (serial and Parallel) for communication with **the outside real-world devices** such as printers, teletypes, telephones, security devices, security monitoring, etc.

These same people say they like a SPHERE COMPUTER SYSTEM because of the **complete COMPUTING POWER** that the **SYSTEM** has to offer.

The advantage of a SPHERE SYSTEM is that in ONE package you can get all the fundamental Computer Features that will allow you to input your programs through the keyboard, process your information, and produce meaningful output.

ALL YOU NEED for your desired computing power FROM ONE SUPPLIER.

COMPATABLE, RELIABLE, POWERFUL.

We have at SPHERE even more options for even greater computing power. See the price list for the details.

As we were conversing with you we found that there are about four general categories of computer users. The categories and the SPHERE SYSTEMS developed to accommodate them are the following:

**KIT BUILDER:** SPHERE SYSTEM 310: you are the one who wants to put a computer together to learn all about it and save some money too.

**HOBBIEST:** SPHERE SYSTEM 320: the one who wants a spare time challenge with ample programming power to do fun, constructive things.

**PROGRAMMER:** SPHERE SYSTEM 330: you are interested in programming for fun or profit. You receive full language, and complete hardware. No kidding around for you.

**DATA HANDLER:** SPHERE SYSTEM 340: to store receive and process mass amounts of information. You mean business.

SPHERE
CORPORATION
791 South 500 West Bountiful
Utah 84010 (801) 292-8466

**Please Contact One Of Our Distributors—**

**Computer Way**—Huntington Beach, California
**Bargain Electronics**—LaMeda, California
**Comput-O-Mat Systems**—Rye, New York
**The Computer Workshop, Inc.**—Montgomery County, Maryland
**Computer Mart Corporated**—Boston, Massachusetts
**the Computer Mart of New York**—New York City, New York
**Communicacions S.A.**—San Jose, Costa Rica
**Computer Country**—Denver, Colorado